Elektronik.
Wir bauen die Elemente.
VALVO

Mikroprozessoren, Mikrocomputer und Peripherieschaltungen
1982

Datenbuch

Elektronik. Wir bauen die Elemente.

Unser Arbeitsgebiet – besonders die Mikroelektronik – entwickelt sich immer rascher zum Motor für eine Vielzahl von Innovationen. Mit gründlicher Information und sorgfältiger Beratung möchten wir Ihnen helfen, diese Entwicklung zu nutzen, um im Wettbewerb vorn zu sein.

Zugegeben, wir sind dabei in einer besonders günstigen Lage: Als Unternehmensbereich Bauelemente des Hauses Philips verbindet Valvo die Erfahrung und Beweglichkeit des deutschen Spezialisten mit der Stärke des weltweit größten Anbieters von elektronischen Bauelementen.

Die Vorteile zeigen sich zum Beispiel in der hohen Innovationsrate, da wir die eigene Forschung und Entwicklung durch internationalen Forschungsverbund ergänzen. Zugleich verfügen wir über das breiteste Produktprogramm in Deutschland. Wir können daher unseren Partnern innovative, vielseitige Problemlösungen aus einer Hand anbieten. Mit Produkten, die pünktlich zur Stelle sind. Hohe Lieferzuverlässigkeit, weit entwickelte Fertigungsverfahren, kompromißlose Qualitätssicherung sind für uns selbstverständlich.

Wie der Erfolg zeigt, ist das eine gute Plattform für die Zusammenarbeit. Damit daraus eine langfristige, erfreuliche Partnerschaft wird, sind wir bereit, schnell zu helfen und Probleme flexibel und unbürokratisch zu lösen.

Information ist der erste Schritt. Sprechen Sie mit uns, wenn es um Bauelemente geht.

Die Stichwortliste gibt einen groben Überblick über unser Vertriebsprogramm, das insgesamt Bauelemente aus hundert Technologien bietet.

Vertriebsprogramm:

- **Integrierte Schaltungen**
  - **Bipolar analog**
  - **Bipolar digital**
  - **MOS**
  - **Hybrid**
- **Mikroprozessoren und -computer**
- **Diskrete Halbleiter**
- **Optoelektronische Bauelemente**
- **Sensoren**
- **Kondensatoren**
- **Widerstände und Potentiometer**
- **Heiß- und Kaltleiter**
- **Hart- und weichmagnetische Ferrite**
- **Fernsehbildröhren und Ablenkmittel**
- **Monitorröhren und Ablenkmittel**
- **Transformatoren**
- **Tuner**
- **Lautsprecher**
- **Spezialröhren**
- **Quarze**
- **Steckverbinder**
- **Leiterplatten**
- **Motoren**

VALVO

Modernste Fertigungseinrichtungen - wie dieser Chip-Bonder - sichern höchste Qualität, Wirtschaftlichkeit und Liefertreue.

Elektronik.
Wir bauen die Elemente.
VALVO

November 1982

Valvo Datenbuch
Mikroprozessoren, Mikrocomputer und Peripherieschaltungen 1982

Sie erhalten hiermit ein Datenbuch über alle unsere Mikroprozessoren, Mikrocomputer und Peripherieschaltungen, die heute in unserem Programm sind, und wir hoffen, Ihnen hiermit ein wichtiges Hilfsmittel für Ihre Arbeit zur Verfügung zu stellen.

Noch einige Hinweise zu diesem Datenbuch:
In einem solchen Mikroprozessor- und Mikrocomputer-Datenbuch können nicht alle mit diesen Bauelementen zusammenarbeitenden peripheren Schaltungen enthalten sein, wie solche mit der I^2C-Schnittstelle, um nur ein Beispiel zu nennen,
ebenfalls ist es nicht möglich, die zu Mikroprozessoren und Mikrocomputern notwendigen Entwicklungssysteme und entsprechende Software aufzuführen und zu beschreiben.
Wünschen Sie zu diesen beiden Themen unsere Unterlagen, so fordern Sie diese bitte gesondert bei uns an.

Mit freundlicher Empfehlung
Valvo
Unternehmensbereich Bauelemente der Philips GmbH

Produktbereich
Mikroprozessoren

Losch

Abteilung DWE

Dahlwitz

Form. 400 014 2.82

Valvo Unternehmensbereich Bauelemente der Philips GmbH, Burchardstr. 19, 2000 Hamburg 1, Telex 2 15 401-0 va d, Telefon (0 40) 32 96-0
Geschäftsführung: Hermanus Koning (Vors.), Dieter Bonow, Karl-Heinz Busacker, Dr. Gert Lorenz, Josef Möllers, Jacob H. Schakel, Prof. Dr. Karl Joachim Schmidt-Tiedemann, Harald Uhl, Reimer Witt
Vors. des Aufsichtsrates: Dr. h.c. Alwin Münchmeyer; Sitz der Ges.: Hamburg, Registergericht Hamburg: 66 HRB 6836

VALVO

Mikroprozessoren, Mikrocomputer und Peripherieschaltungen
1982

Datenbuch

Herausgeber: Valvo
Unternehmensbereich Bauelemente der Philips GmbH
Burchardstraße 19, 2000 Hamburg 1

Verlag Boysen + Maasch, Hermannstraße 31, 2000 Hamburg 1

ISBN 3-87095-165-6

Druck: Photo Copie GmbH, 2000 Hamburg 1

September 1982

Contents

Section 1. Data Communications

SC2651 Programmable Communications Interface (PCI) 1-2
SC2652 Multi-Protocol Communications Controller (MPCC) 1-16
SC2653 Polynomial Generator Checker (PGC) 1-34
SC2661 Enhanced Programmable Communications Interface (EPCI) 1-52
SC2661 24-pin Enhanced Programmable Communications Interface (EPCI) 1-87
App Note 402: 2661 Operating Mode Switching Procedures 1-90
SC2681 Dual Asynchronous Receiver/Transmitter (DUART) 1-69

Section 2. Video Display

SC2670 Display Character and Graphics Generator (DCGG) 2-2
SC2671 Programmable Keyboard & Comm Controller (PKCC) 2-15
SC2672 Programmable Video Timing Controller (PVTC) 2-36
SC2673 Video Attributes Controller (VAC) 2-57

Section 3. Single Chip 8-Bit Microcomputers

MAB8021 Single Chip 8-Bit Microcomputer 3-3
MAB8041A Single Chip 8-Bit Microcomputer 3-21
MAB8048H MAB8035HL Single Chip 8-Bit Microcomputers 3-23
MAB8049H MAB8039H Single Chip 8-Bit Microcomputers 3-55
MAB8050H MAB8040H Single Chip 8-Bit Microcomputers 3-87
MAB8400 - Family Single Chip 8-Bit Microcomputers 3-117

Section 4. SC68000 16-Bit Microprocessor Family

SC68000 16-Bit Microprocessor 4-2
SC68230 Parallel Interface/Timer 4-51
***SC68430** Direct Memory Access Interface (DMAI) 4-81
***SC68451** Memory Management Unit 4-82
SC68681 Dual Asynchronous Receiver/Transmitter (DUART) 4-83

Section 5. Video Game

***MAB2650A** Microprocessor 5-2
***MEB2621/22** Universal Sync Generator (USG) 5-6
***MEB2636** Programmable Video Interface (PVI) 5-7

*PRODUCT BRIEF

Section 6. Bipolar 8-Bit Microprocessor Family

8X300 Micro Controller 6-4
8X305 Micro Controller 6-24
8X310 Interrupt Controller 6-49
8X320 Bus Interface Array, 2-port RAM for 8/16-bit mailbox interfacing 6-50
8X330 Floppy Disk Controller 6-59
8X350 Bipolar RAM, 256 x 8 high speed memory with bus interface 6-75
8X353 Bipolar RAM, 32 x 8 high speed memory with bus interface 6-79
8X355 LIFO Memory, 32 x 8 high speed LIFO stack with bus interface 6-80
8X360 Memory Address Director 6-81
8X371 Transparent I/O Port, 8-bit bidirectional 6-82
8X372 Addressable I/O Port, 8-bit bidirectional, synchronous 6-89
8X374 Addressable I/O Port, 8-bit bidirectional, synchronous, with parity 6-98
8X376 Addressable I/O Port, 8-bit bidirectional, asynchronous 6-89
8X382 Addressable I/O Port, 4-in/4-out 6-99

8T31 Transparent I/O Port, 8-bit bidirectional 6-109
8X31 Transparent I/O Port, 8-bit bidirectional 6-109
8T32 Addressable I/O Port, 8-bit bidirectional, synchronous 6-113
8X32 Addressable I/O Port, 8-bit bidirectional, synchronous 6-113
8T33 Addressable I/O Port, 8-bit bidirectional, synchronous 6-113
8T35 Addressable I/O Port, 8-bit bidirectional, asynchronous 6-113
8T36 Addressable I/O Port, 8-bit bidirectional, asynchronous 6-113
8X36 Addressable I/O Port, 8-bit bidirectional, asynchronous 6-113
8T39 Addressable Bus Expander 6-121
8X42 Addressable I/O Port, 4-in/4-out 6-113
8T58 Transparent Bus Expander 6-125

82HS137 Bipolar PROM, 1024 x 4 6-129
82HS147 Bipolar PROM, 512 x 8 6-131
82HS181 Bipolar PROM, 1024 x 8 6-133
82HS185 Bipolar PROM, 2048 x 4 6-136
82HS191 Bipolar PROM, 2048 x 8 6-138
82HS195 Bipolar PROM, 4096 x 4 6-141

Section 7. Bipolar Bit Slice Microprocessors

N3001 Microprogram Control Unit .... } data sheets
N3002 Central Processing Element .... } on request

Section 8. Package Outlines 8-1

PART NUMBER CONVERSION NOTICE

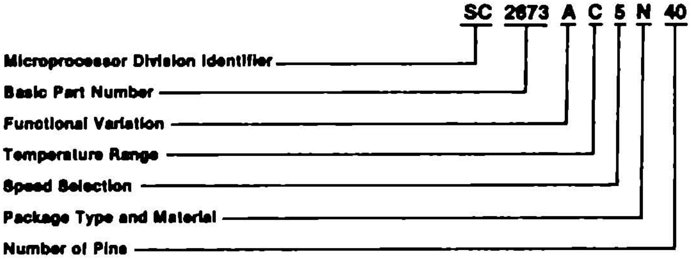
New part numbers have been assigned to products contained in this data manual. Conversion from old part numbers to new part numbers is shown below.

NEW NUMBER	OLD NUMBER
MAB2650AD	2650AI
MAB2650AD-1	2650A-1I
MAB2650AP	2650AN
MAB2650AP-1	2650A-1N
MEB2621P	2621N
MEB2636P	2636N
SC2651CSI28	2651I
SC2651CSN28	2651N
SC2652C1I40	2652I
SC2652C1N40	2652N
SC2652C2I40	2652-1I
SC2652C2N40	2652-1N
SC2653CSI16	2653I
SC2653CSN16	2653N
SC2661ACSI28	2661-1I
SC2661BCSI28	2661-2I
SC2661CCSI28	2661-3I
SC2661ACSN28	2661-1N
SC2661BCSN28	2661-2N
SC2661CCSN28	2661-3N
SC2670•CSI28	2670I
SC2670•CSN28	2670N
SC2671ACSI40	2671I
SC2671ACSN40	2671N
SC2672C4I40	2672I
SC2672C4N40	2672N
SC2673BC5I40	2673I
SC2673BC5N40	2673N
SC2673AC5I40	2673AI
SC2673AC5N40	2673AN

• See Data Sheet

PART NUMBERING SYSTEM

The following example illustrates the meaning of the various fields in the part number:


SC 2673 A C 5 N 40
Microprocessor Division Identifier
Basic Part Number
Functional Variation
Temperature Range
Speed Selection
Package Type and Material
Number of Pins

DEFINITION OF TERMS

Data Sheet Identification	Product Status	Definition
Preview	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Advance Information	Sampling or Pre-Production	This data sheet contains advance information and specifications are subject to change without notice.
Preliminary	First Production	This data sheet contains preliminary data and supplementary data will be published at a later date. Valvo reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. Valvo reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Valvo reserves the right to make changes in the products contained in this book in order to improve design or performance and to supply the best possible products. Valvo also assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representations that the circuits are free from patent infringement. Applications for any integrated circuits contained in this publication are for illustration purposes only and Valvo makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification. Reproduction of any portion hereof without the prior written consent of Valvo is prohibited.

Section 1
Data Communications

VALVO

JANUARY 1982

PROGRAMMABLE COMMUNICATIONS INTERFACE (PCI) SC2651

DESCRIPTION

The Signetics 2651 PCI is a universal synchronous/asychronous data communications controller chip designed for microcomputer systems. It interfaces directly to the Signetics 2650 microprocessor and may be used in a polled or interrupt driven system environment. The 2651 accepts programmed instructions from the microprocessor and supports many serial data communication disciplines, synchronous and asynchronous, in the full or half-duplex mode.

The PCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The 2651 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode.

The PCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

FEATURES

- **Synchronous operation**
  - **5 to 8-bit characters**
  - **Single or double SYN operation**
  - **Internal character synchronization**
  - **Transparent or non-transparent mode**
  - **Automatic SYN or DLE-SYN insertion**
  - **SYN or DLE stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**

- **Asynchronous operation**
  - **5 to 8-bit characters**
  - **1, 1 1/2 or 2 stop bits**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
    dc to 62.5K bps (16X clock)
    dc to 15.625K bps (64X clock)

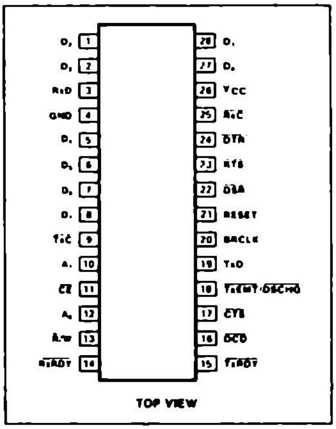
OTHER FEATURES

- **Internal or external baud rate clock**
- **16 internal rates-50 to 19,200 baud**
- **Double buffered transmitter and receiver**
- **Full or half duplex operation**
- **Fully compatible with 2650 CPU**
- **TTL compatible inputs and outputs**
- **Single 5V power supply**
- **No system clock required**
- **28-pin dual in-line package**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front end processors**
- **Remote data concentrators**
- **Computer to computer links**
- **Serial peripherals**

PIN CONFIGURATION



PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27,28,1,2, 5-8	D_0-D_7	8-bit data bus	I/O
21	RESET	Reset	I
12,10	A_0-A_1	Internal register select lines	I
13	$\overline{R}/W$	Read or write command	I
11	$\overline{CE}$	Chip enable input	I
22	$\overline{DSR}$	Data set ready	I
24	$\overline{DTR}$	Data terminal ready	O
23	$\overline{RTS}$	Request to send	O
17	$\overline{CTS}$	Clear to send	I
16	$\overline{DCD}$	Data carrier detected	I
18	$\overline{TxEMT/DSCHG}$	Transmitter empty or data set change	O
9	$\overline{TxC}$	Transmitter clock	I/O
25	$\overline{RxC}$	Receiver clock	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	$\overline{TxRDY}$	Transmitter ready	O
14	$\overline{RxRDY}$	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	Vcc	+5V supply	I
4	GND	Ground	I

ORDERING CODE

PACKAGES	COMMERCIAL RANGE V_{CC} = 5V ±5%, T_A = 0°C to 70°C
Plastic DIP	SC2651CSN28
Ceramic DIP	SC2651CSI28

SEE 2661 FOR ENHANCED PART.

BAUD RATE	THEORETICAL FREQUENCY 16X CLOCK	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
50	0.8 KHz	0.8 KHz	--	6336
75	1.2	1.2	—	4224
110	1.76	1.76	—	2880
134.5	2.152	2.1523	0.016	2355
150	2.4	2.4	—	2112
300	4.8	4.8	--	1056
600	9.6	9.6	—	528
1200	19.2	19.2	—	264
1800	28.8	28.8	—	176
2000	32.0	32.081	0.253	158
2400	38.4	38.4	—	132
3600	57.6	57.6	—	88
4800	76.8	76.8	—	66
7200	115.2	115.2	—	44
9600	153.6	153.6	—	33
19200*	307.2	316.8	3.125	16

NOTE
*Error at 19200 can be reduced to zero by using crystal frequency 4.9152MHz
16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X

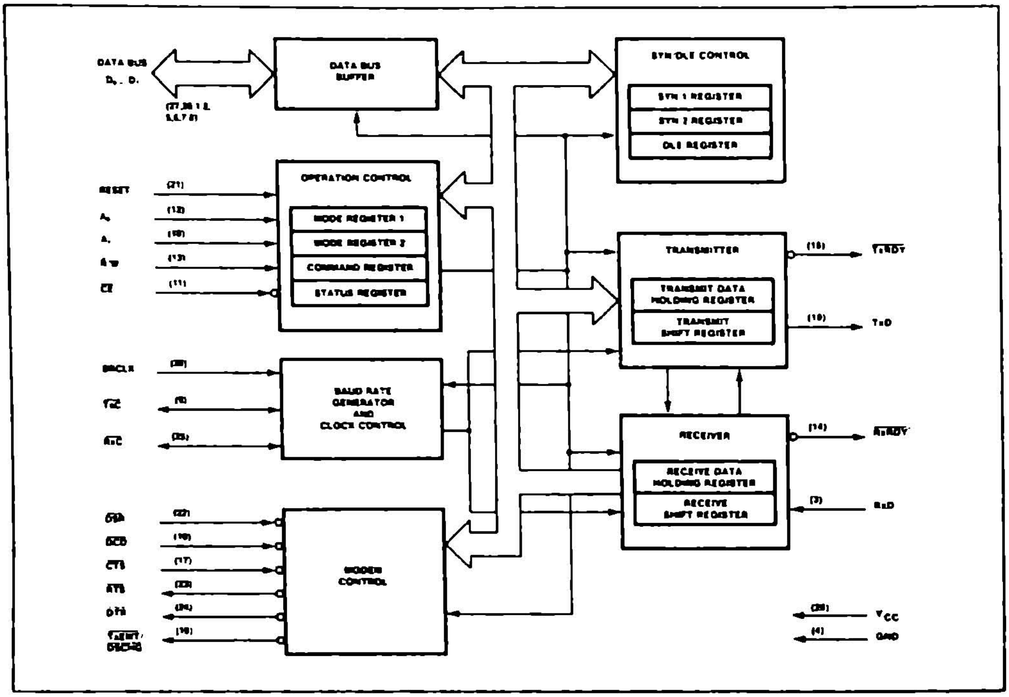
Table 1 BAUD RATE GENERATOR CHARACTERISTICS
Crystal Frequency = 5.0688MHz

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
Vcc	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the 2651. This signal asynchronously terminates any device activity and clears the Mode, Command and Status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	10,12	I	Address lines used to select internal PCI registers.
$\overline{R}$/W	13	I	Read command when low, write command when high.
$\overline{CE}$	11	I	Chip enable command. When low, indicates that control and data lines to the PCI are valid and that the operation specified by the $\overline{R}$/W, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the tri-state condition.
D_7-D_0	8,7,6,5, 2,1,28,27	I/O	8-bit, three-state data bus used to transfer commands, data and status between PCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{TxRDY}$	15	O	This output is the complement of Status Register bit SR0. When low, it indicates that the Transmit Data Holding Register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{RxRDY}$	14	O	This output is the complement of Status Register bit SR1. When low, it indicates that the Receive Data Holding Register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{TxEMT}/\overline{DSCHG}$	18	O	This output is the complement of Status Register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. This output goes high when the Status Register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

Table 2 CPU-RELATED SIGNALS

PROGRAMMABLE COMMUNICATIONS INTERFACE (PCI) 2651

BLOCK DIAGRAM


DATA BUS
D0 – D7
(27,28,1,2, 5,6,7,8)
DATA BUS BUFFER
SYN/DLE CONTROL
SYN 1 REGISTER
SYN 2 REGISTER
DLE REGISTER
OPERATION CONTROL
MODE REGISTER 1
MODE REGISTER 2
COMMAND REGISTER
STATUS REGISTER
RESET (21)
A0 (12)
A1 (10)
R/W (13)
CE (11)
TRANSMITTER
TRANSMIT DATA HOLDING REGISTER
TRANSMIT SHIFT REGISTER
TxRDY (15)
TxD (19)
BRCLK (20)
TxC (9)
RxC (25)
BAUD RATE GENERATOR AND CLOCK CONTROL
RECEIVER
RECEIVE DATA HOLDING REGISTER
RECEIVE SHIFT REGISTER
RxRDY (14)
RxD (3)
DSR (22)
DCD (16)
CTS (17)
RTS (23)
DTR (24)
TxEMT/DSCHG (18)
MODEM CONTROL
VCC (26)
GND (4)

BLOCK DIAGRAM

The PCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains Mode Registers 1 and 2, the Command Register, and the Status Register. Details of register addressing and protocol are presented in the PCI Programming section of this data sheet.

Timing

The PCI contains a Baud Rate Generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See Table 1.

Receiver

The Receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The Transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

INTERFACE SIGNALS

The PCI interface signals can be grouped into two types: the CPU-related signals (shown in Table 2), which interface the 2651 to the microprocessor system, and the device-related signals (shown in Table 3), which are used to interface to the communications device or system.

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
BRCLK	20	I	5.0688MHz clock input to the internal baud rate generator. Not required if external receiver and transmitter clocks are used
$\overline{\text{RxC}}$	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by Mode Register 1. Data is sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin becomes an output at 1X the programmed baud rate.*
$\overline{\text{TxC}}$	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by Mode Register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin becomes an output at 1X the programmed baud rate.*
RxD	3	I	Serial data input to the receiver. "Mark" is high, "Space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "Space" is low. Held in Mark condition when the transmitter is disabled.
$\overline{\text{DSR}}$	22	I	General purpose input which can be used for Data Set Ready or Ring Indicator condition. Its complement appears as Status Register bit SR7. Causes a low output on TxEMT/$\overline{\text{DSCHG}}$ when its state changes.
$\overline{\text{DCD}}$	16	I	Data Carrier Detect input. Must be low in order for the receiver to operate. Its complement appears as Status Register bit SR6. Causes a low output on TxEMT/$\overline{\text{DSCHG}}$ when its state changes.
$\overline{\text{CTS}}$	17	I	Clear to Send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the Transmit Shift Register will be transmitted before termination.
$\overline{\text{DTR}}$	24	O	General purpose output which is the complement of Command Register bit CR1. Normally used to indicate Data Terminal Ready.
$\overline{\text{RTS}}$	23	O	General purpose output which is the complement of Command Register bit CR5. Normally used to indicate Request to Send.

NOTE
*$\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ outputs have short circuit protection max. C_L 100pf

Table 3 DEVICE-RELATED SIGNALS

OPERATION

The functional operation of the 2651 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the PCI Programming section of this data sheet.

After programming, the PCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2651 is conditioned to receive data when the $\overline{\text{DCD}}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and the stop bit(s) have been assembled. The data is then transferred to the Receive Data Holding Register, the RxRDY bit in the status register is set, and the $\overline{\text{RxRDY}}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the Holding Register are set to zero. The Parity Error, Framing Error, and Overrun Error status bits are strobed into the status register on the positive going edge of $\overline{\text{RxC}}$ corresponding to the received character boundary. If a break condition is detected (RxD is low for the entire character as well as the stop bit(s)), only one character consisting of all zeros (with the FE status bit set) will be transferred to the Holding Register. The RxD input must return to a high condition before a search for the next start bit begins.

When the PCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN (CR2). In this mode, as data is shifted into the Receiver Shift Register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the PCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization). When synchronization has been achieved, the PCI continues to assemble characters and transfer them to the Holding Register, setting the RxRDY status bit and asserting the $\overline{\text{RxRDY}}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the Holding Register. Note that the SYN characters used to establish initial synchronization are not transferred to the Holding Register in any case.

Transmitter

The PCI is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The 2651 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the Transmit Data Holding Register, these conditions are negated. Data is transferred from the Holding Register to the Transmit Shift Register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

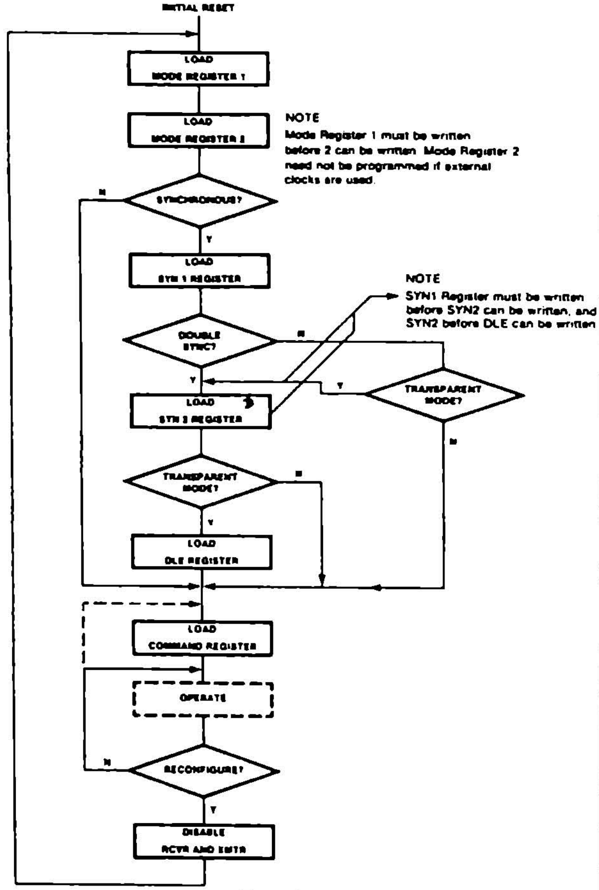
In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the Transmit Holding Register, the TxD output remains in the marking (high) condition and the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the Holding Register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the Send Break command bit high.

In the synchronous mode, when the 2651 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the PCI unless the CPU fails to send a new character to the PCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the PCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the Transmit Data Holding Register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in THR.

PCI PROGRAMMING

Prior to initiating data communications, the 2651 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The PCI can be reconfigured at any time during program execution. However, if the change has an effect on the reception of a character the receiver should be disabled. Alternatively if

2651 INITIALIZATION FLOW CHART



NOTE
Mode Register 1 must be written before 2 can be written. Mode Register 2 need not be programmed if external clocks are used.

NOTE
SYN1 Register must be written before SYN2 can be written, and SYN2 before DLE can be written

Figure 1

$\overline{CE}$	A_1	A_0	$\overline{R}$/W	FUNCTION
1	X	X	X	Tri-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers 1/2
0	1	0	1	Write mode registers 1/2
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC Characteristics section for timing requirements.

Table 4 2651 REGISTER ADDRESSING

the change is made $1\frac{1}{2}$ RxC periods after RxRDY goes active it will affect the next character assembly. A flowchart of the initialization process appears in Figure 1.

The internal registers of the PCI are accessed by applying specific signals to the $\overline{CE}$, $R/\overline{W}$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in Table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and $R/\overline{W} = 1$. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses Mode Register 1, and a subsequent operation addresses Mode Register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 Register and Mode Register 1 by a RESET input or by performing a "Read Command Register" operation, but are unaffected by any other read or write operation.

The 2651 register formats are summarized in Tables 5, 6, 7 and 8. Mode Registers 1 and 2 define the general operational characteristics of the PCI, while the Command Register controls the operation within this basic frame-work. The PCI indicates its status in the Status Register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asychronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit). In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN Detect, but the normal synchronization sequence is used. Also DLE stripping and DLE Detect (with MR14 = 0) are enabled.

Mode Register 2 (MR2)

Table 6 illustrates Mode Register 2. MR23, MR22, MR21, and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable. When driven by a 5.0688 MHz input at the BRCLK input (pin 20), the BRG output has zero error except at 134.5, 2000, and 19,200 baud, which have errors of +0.016%, +0.235%, and +3.125% respectively.

MR25 and MR24 select either the BRG or the external inputs $\overline{TxC}$ and $\overline{RxC}$ as the clock source for the transmitter and receiver, respectively. If the BRG clock is selected,

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
		Parity Type	Parity Control	Character Length		Mode and Baud Rate Factor	
ASYNCH: STOP BIT LENGTH 00 = INVALID 01 = 1 STOP BIT 10 = $1\frac{1}{2}$ STOP BITS 11 = 2 STOP BITS		0 = ODD 1 = EVEN	0 = DISABLED 1 = ENABLED	00 = 5 BITS 01 = 6 BITS 10 = 7 BITS 11 = 8 BITS		00 = SYNCHRONOUS 1X RATE 01 = ASYNCHRONOUS 1X RATE 10 = ASYNCHRONOUS 16X RATE 11 = ASYNCHRONOUS 64X RATE	
SYNCH: NUMBER OF SYN CHAR 0 = DOUBLE SYN 1 = SINGLE SYN	SYNCH: TRANSPARENCY CONTROL 0 = NORMAL 1 = TRANSPARENT						

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 5 MODE REGISTER 1 (MR1)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	Receiver Clock	Baud Rate Selection			
NOT USED		0 = EXTERNAL 1 = INTERNAL	0 = EXTERNAL 1 = INTERNAL	0000 = 50 BAUD 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 BAUD 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19,200	

Table 6 MODE REGISTER 2 (MR2)

the baud rate factor in asynchronous mode is 16X regardless of the factor selected by MR11 and MR10. In addition, the corresponding clock pin provides an output at 1X the baud rate

Command Register (CR)

Table 7 illustrates Command Register Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{\mathrm{RxC}}$ rising edge Disabling the receiver causes $\overline{\mathrm{RxRDY}}$ to go high (inactive) If the transmitter is disabled, it will complete the transmission of the character in the Transmit Shift Register (if any) prior to terminating operation The TxD output will then remain in the marking state (high) while the $\overline{\mathrm{TxRDY}}$ and $\overline{\mathrm{TxEMT}}$ will go high (inactive) If the receiver is disabled, it will terminate operation immediately Any character being assembled will be neglected

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{\mathrm{DTR}}$ and $\overline{\mathrm{RTS}}$ outputs Data at the outputs is the logical complement of the register data

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character Normal operation resumes when CR3 is cleared The TxD line will go high for a least one bit time before beginning transmission of the next character in the Transmit Data Holding Register In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the Transmit Data Holding Register. CR3 should be reset in response to the next TxRDY

Setting CR4 causes the error flags in the Status Register (SR3, SR4, and SR5) to be cleared This is a one time command There is no internal latch for this bit

The PCI can operate in one of four submodes within each major mode (synchronous or asynchronous) The operational sub-mode is determined by CR7 and CR6 CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the Mode and Status Register instructions

In asynchronous mode, CR7-CR6 = 01 places the PCI in the Automatic Echo mode Clocked, regenerated received data is automatically directed to the TxD line while normal receiver operation continues The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled CPU to receiver communications continues normally, but the CPU to transmitter link is disabled Only the first character of a break condition is echoed The TxD output will go high until the next valid start is detected. The following conditions are true while in Automatic Echo mode

1 Data assembled by the receiver is automatically placed in the Transmit Holding Register and retransmitted by the transmitter on the TxD output
2 The transmitter is clocked by the receive clock
3 $\overline{\mathrm{TxRDY}}$ output = 1
4 The $\overline{\mathrm{TxEMT}}/\overline{\mathrm{DSCHG}}$ pin will reflect only the data set change condition
5 The TxEN command (CR0) is ignored

In synchronous mode, CR7-CR6 = 01 places the PCI in the Automatic SYN/DLE Stripping mode The exact action taken depends on the setting of bits MR17 and MR16

1 In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the Receive Data Holding Register (RHR)
2 In the non-transparent, double SYN mode (MR17-MR16 = 00) characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR However only the first SYN1 of an SYN1-SYN1 pair is stripped
3 In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR However only the first DLE of a DLE-DLE pair is stripped

Note that Automatic Stripping mode does not affect the setting of the DLE Detect and SYN Detect status bits (SR3 and SR5)

Two diagnostic sub-modes can also be configured In Local Loop Back mode (CR7-CR6 = 10), the following loops are connected internally

1 The transmitter output is connected to the receiver input
2 $\overline{\mathrm{DTR}}$ is connected to $\overline{\mathrm{DCD}}$ and $\overline{\mathrm{RTS}}$ is connected to $\overline{\mathrm{CTS}}$
3 The receiver is clocked by the transmit clock
4 The $\overline{\mathrm{DTR}}$, $\overline{\mathrm{RTS}}$ and $\overline{\mathrm{TxD}}$ outputs are held high
5 The $\overline{\mathrm{CTS}}$, $\overline{\mathrm{DCD}}$, $\overline{\mathrm{DSR}}$ and RxD inputs are ignored

Additional requirements to operate in the Local Loop Back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1 CR2 (RxEN) is ignored by the PCI

The second diagnostic mode is the Remote Loop Back mode (CR7-CR6 = 11). In this mode

1 Data assembled by the receiver is automatically placed in the Transmit Holding Register and retransmitted by the transmitter on the TxD output
2 The transmitter is clocked by the receive clock
3 No data is sent to the local CPU, but the error status conditions (PE, OE, FE) are set
4. The $\overline{\mathrm{RxRDY}}$, $\overline{\mathrm{TxRDY}}$, and $\overline{\mathrm{TxEMT}}/\overline{\mathrm{DSCHG}}$ outputs are held high
5 CR1 (TxEN) is ignored
6 All other signals operate normally

Status Register

The data contained in the Status Register (as shown in Table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the Transmitter Ready (TxRDY) status bit It, and its corresponding output are valid only when the transmitter is enabled If equal to 0, it indicates that the Transmit Data Holding Register has been loaded by the CPU and the data has not been transferred to the Transmit Shift Register If set equal to 1, it indicates that the Holding Register is ready to accept data from the CPU This bit is initially set when the Transmitter is enabled by CR0, unless a character

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		Request to Send	Reset Error		Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = NORMAL OPERATION 01 = ASYNCH AUTOMATIC ECHO MODE SYNCH SYN AND/OR DLE STRIPPING MODE 10 = LOCAL LOOP BACK 11 = REMOTE LOOP BACK		0 = FORCE $\overline{\mathrm{RTS}}$ OUTPUT HIGH 1 = FORCE $\overline{\mathrm{RTS}}$ OUTPUT LOW	0 = NORMAL 1 = RESET ERROR FLAG IN STATUS REG (FE, OE, PE/DLE DETECT)	ASYNCH: FORCE BREAK 0 = NORMAL 1 = FORCE BREAK SYNCH: SEND DLE 0 = NORMAL 1 = SEND DLE	0 = DISABLE 1 = ENABLE	0 = FORCE $\overline{\mathrm{DTR}}$ OUTPUT HIGH 1 = FORCE $\overline{\mathrm{DTR}}$ OUTPUT LOW	0 = DISABLE 1 = ENABLE

Table 7 COMMAND REGISTER (CR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ INPUT IS HIGH 1 = $\overline{DSR}$ INPUT IS LOW	0 = $\overline{DCD}$ INPUT IS HIGH 1 = $\overline{DCD}$ INPUT IS LOW	ASYNCH: 0 = NORMAL 1 = FRAMING ERROR SYNCH: 0 = NORMAL 1 = SYN CHAR DETECTED	0 = NORMAL 1 = OVERRUN ERROR	ASYNCH: 0 = NORMAL 1 = PARITY ERROR SYNCH: 0 = NORMAL 1 = PARITY ERROR OR DLE CHAR RECEIVED	0 = NORMAL 1 = CHANGE IN $\overline{DSR}$ OR $\overline{DCD}$, OR TRANSMIT SHIFT REGISTER IS EMPTY	0 = RECEIVE HOLDING REG EMPTY 1 = RECEIVE HOLDING REG HAS DATA	0 = TRANSMIT HOLDING REG BUSY 1 = TRANSMIT HOLDING REG EMPTY

Table 8 STATUS REGISTER (SR)

has previously been loaded into the Holding Register. It is not set when the Automatic Echo or Remote Loop Back modes are programmed. When this bit is set, the $\overline{TxRDY}$ output pin is low. In the Automatic Echo and Remote Loop Back modes, the output is held high.

SR1, the Receiver Ready (RxRDY) status bit, indicates the condition of the Receive Data Holding Register. If set, it indicates that a character has been loaded into the Holding Register from the Receive Shift Register and is ready to be read by the CPU. If equal to zero, there is no new character in the Holding Register. This bit is cleared when the CPU reads the Receive Data Holding Register or when the receiver is disabled by CR2. When set, the $\overline{RxRDY}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs or that the Transmit Shift Register has completed transmission of a character and no new character has been loaded into the Transmit Data Holding Register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the Transmit Data Holding Register. The DSCHG condition is enabled when TxEN=1 or RxEN=1. It is cleared when the Status Register is read by the CPU. When SR2 is set, the TxEMT/DSCHG output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching the DLE Register has been received. However, only the first DLE of two successive DLEs will set SR3. This bit is cleared when the receiver is disabled and by the Reset Error command, CR4.

The Overrun Error status bit, SR4, indicates that the previous character loaded into the Receive Holding Register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled and by the Reset Error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by the programmed number of stop bits. (If 1.5 stop bits are programmed, only the first stop bit is checked.) If RHR = 0 when SR5 = 1 a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character is single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the Reset Error command is given in asynchronous mode, and when the Status Register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{DCD}$ and $\overline{DSR}$ inputs respectively. A low input sets its corresponding status bit and a high input clears it.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	$I_{OL} = 1.6mA$			0.4	
V_{OH}	High	$I_{OH} = -100\mu A$	2.4			
I_{IL}	Input leakage current	$V_{IN} = 0$ to $5.25V$	-10		10	μA
Tristate	Output leakage current					μA
I_{LH}	Data bus high	$V_O = 4.0V$	-10		10	
I_{LL}	Data bus low	$V_O = 0.45V$	-10		10	
I_{CC}	Power supply current				150	mA

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	-65 to +150	°C
All voltages with respect to ground[3]	-0.5 to +6.0	V

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Pulse width					ns
t_{RES}	Reset		1000			
t_{CE}	Chip enable		300			
	Setup and hold time					ns
t_{AS}	Address setup		20			
t_{AH}	Address hold		20			
t_{CS}	$\overline{R}$/W control setup		20			
t_{CH}	$\overline{R}$/W control hold		20			
t_{DS}	Data setup for write		225			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	C_L = 100pF			250	ns
t_{DF}	Data bus floating time for read	C_L = 100pF			150	ns
t_{CED}	$\overline{CE}$ to $\overline{CE}$ delay		700			ns
	Input clock frequency					MHz
f_{BRG}	Baud rate generator		1.0	5.0688	5.0738	
$f_{R/T}$[10]	$\overline{TxC}$ or $\overline{RxC}$		dc		1.0	
	Clock width					ns
t_{BRH}[9]	Baud rate high		70			
t_{BRL}[9]	Baud rate low		70			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		500			
$t_{R/TL}$[10]	$\overline{TxC}$ or $\overline{RxC}$ low		500			
t_{TXD}	TxD delay from falling edge of $\overline{TxC}$	C_L = 100pF			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output[8]	C_L = 100pF		0		ns

NOTES

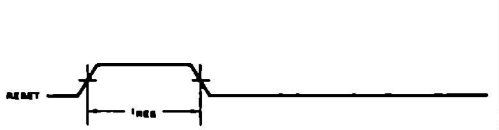
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (IQ ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8V and 2.0V for outputs. Input levels for testing are 0.45V and 2.4V.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. $\overline{TxRDY}$, $\overline{RxRDY}$ and $\overline{TxEMT/DSCHG}$ outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688 MHz f_{BRG}, t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.
10. $f_{R/T}$ and $t_{R/TL}$ shown for all modes except Local Loopback. For Local Loopback mode $f_{R/T}$ = 0.7 MHz and $t_{R/TL}$ = 700ns min.

CAPACITANCE $T_A = 25°C$, $V_{CC} = 0V$

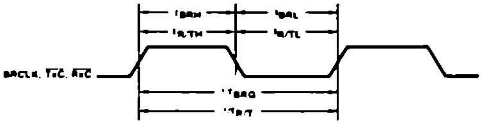
PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Capacitance					pF
C_{IN}	Input	fc = 1MHz Unmeasured pins tied to ground			20	
C_{OUT}	Output				20	
$C_{I/O}$	Input/Output				20	

TIMING DIAGRAMS

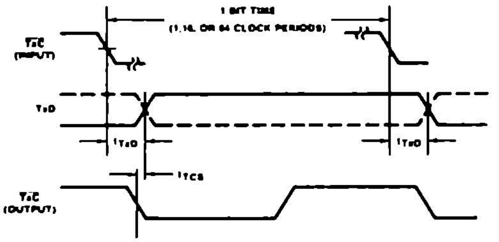
RESET



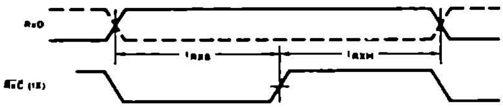
CLOCK


tBRH
tBRL
tR/TH
tR/TL
BRCLK, TxC, RxC
tBRC
tR/T

TRANSMIT

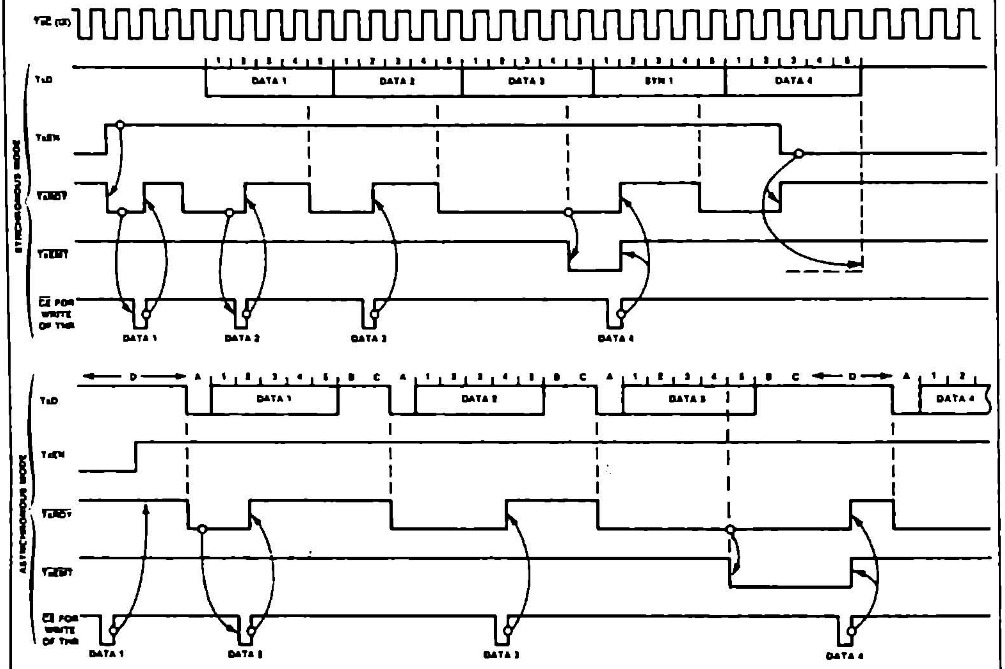

1 BIT TIME
(1, 16, OR 64 CLOCK PERIODS)
TxC (INPUT)
TxD
tTxD
tTxD
tTCS
TxC (OUTPUT)

RECEIVE


RxD
tRXS
tRXH
RxC (1X)

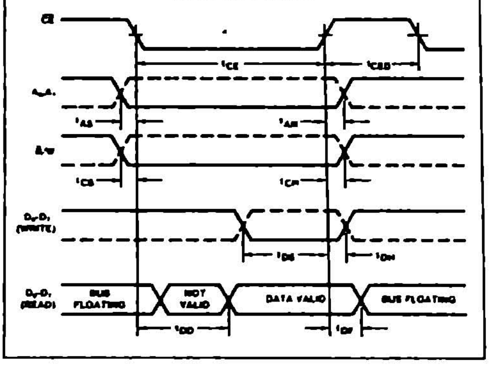
TIMING DIAGRAMS (Cont'd)

TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits (in asynchronous mode))


TxC (1X)
TxD
DATA 1
DATA 2
DATA 3
SYN 1
DATA 4
TxEN
TxRDY
TxEMT
CE FOR WRITE OF THR
SYNCHRONOUS MODE
ASYNCHRONOUS MODE
A
B
C
D
1
2
3
4
5

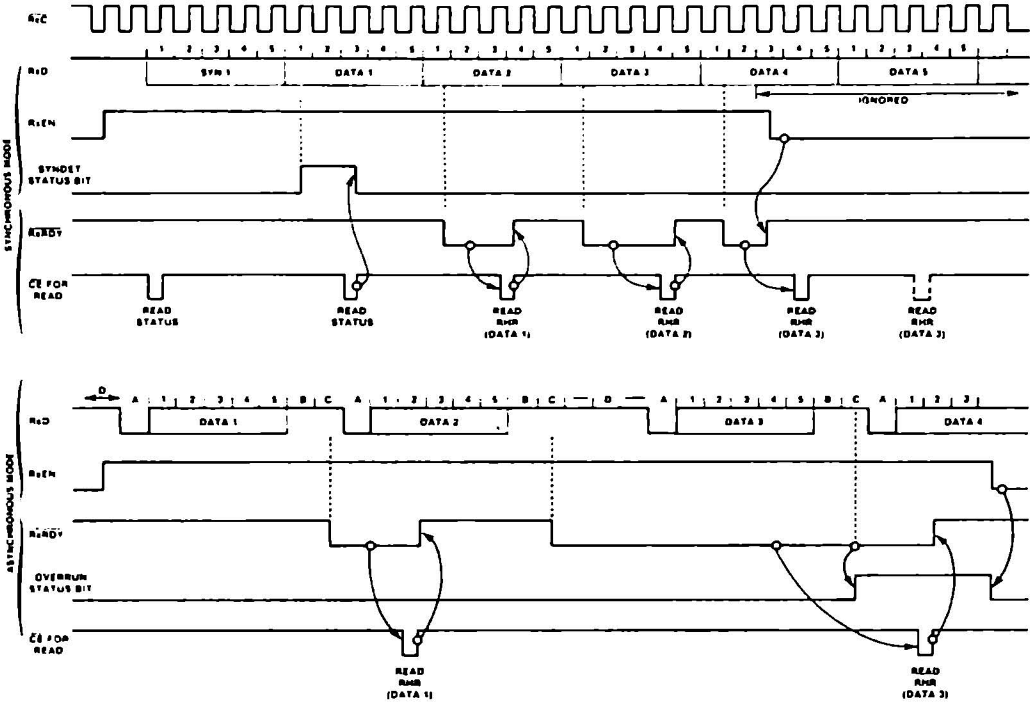
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit

READ AND WRITE


CE
A_0-A_1
R/W
D_0-D_7 (WRITE)
D_0-D_7 (READ)
t_{CE}
t_{CED}
t_{AS}
t_{AH}
t_{CS}
t_{CH}
t_{DS}
t_{DH}
t_{DD}
t_{DF}
BUS FLOATING
NOT VALID
DATA VALID
BUS FLOATING

TIMING DIAGRAMS (Cont'd)

RxRDY (Shown for 5-bit characters, no parity, 2 stop bits (in asynchronous mode).)

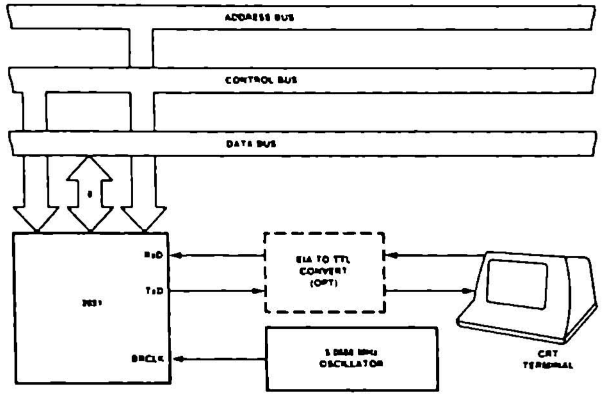

RxC
RxD
1 2 3 4 5
SYN 1
DATA 1
DATA 2
DATA 3
DATA 4
DATA 5
IGNORED
RxEN
SYNDET STATUS BIT
RxRDY
CE FOR READ
READ STATUS
READ STATUS
READ RHR (DATA 1)
READ RHR (DATA 2)
READ RHR (DATA 3)
READ RHR (DATA 3)
SYNCHRONOUS MODE
D
A
B
C
RxD
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
RxRDY
OVERRUN STATUS BIT
CE FOR READ
READ RHR (DATA 1)
READ RHR (DATA 3)
ASYNCHRONOUS MODE

NOTES

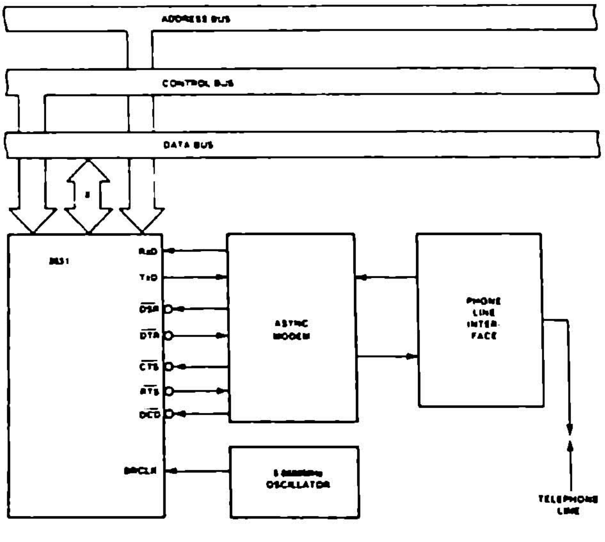
A – Start bit
B – Stop bit 1
C – Stop bit 2
D – TxD marking condition

TYPICAL APPLICATIONS

ASYNCHRONOUS INTERFACE TO CRT TERMINAL

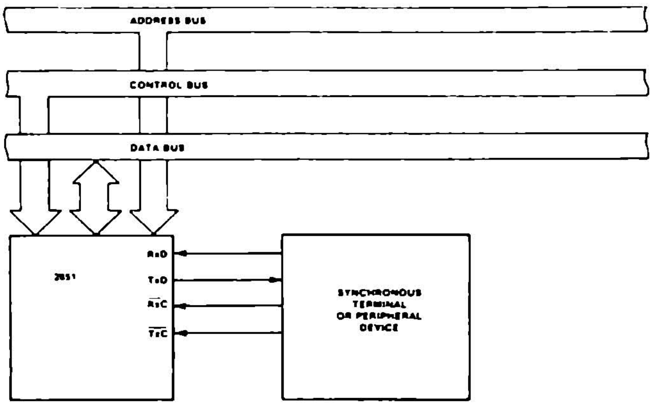

ADDRESS BUS
CONTROL BUS
DATA BUS
8
2651
RxD
TxD
BRCLK
EIA TO TTL CONVERT (OPT)
5.0688 MHz OSCILLATOR
CRT TERMINAL

ASYNCHRONOUS INTERFACE TO TELEPHONE LINES

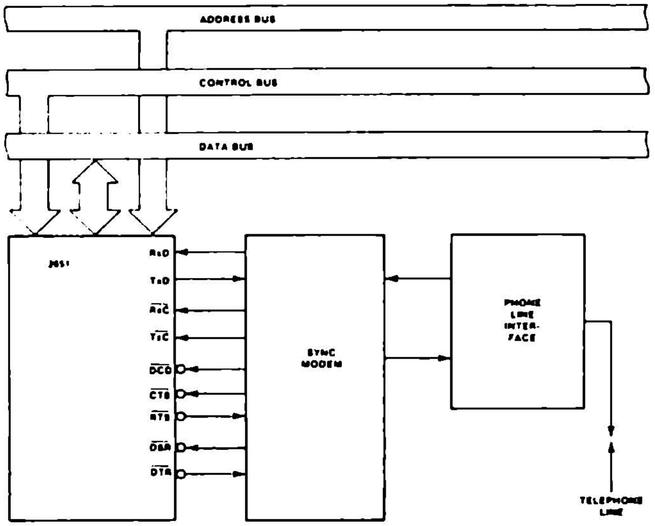

ADDRESS BUS
CONTROL BUS
DATA BUS
8
2651
RxD
TxD
DSR
DTR
CTS
RTS
DCD
BRCLK
ASYNC MODEM
PHONE LINE INTER-FACE
5.0688MHz OSCILLATOR
TELEPHONE LINE

TYPICAL APPLICATIONS (Cont'd)

SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE


ADDRESS BUS
CONTROL BUS
DATA BUS
2651
RxD
TxD
RxC
TxC
SYNCHRONOUS TERMINAL OR PERIPHERAL DEVICE

SYNCHRONOUS INTERFACE TO TELEPHONE LINES


ADDRESS BUS
CONTROL BUS
DATA BUS
2651
RxD
TxD
RxC
TxC
DCD
CTS
RTS
DSR
DTR
SYNC MODEM
PHONE LINE INTER-FACE
TELEPHONE LINE

Manufacturer reserves the right to make design and process changes and improvements.

JANUARY 1982

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER (MPCC) SC2652

DESCRIPTION

The 2652 Multi-Protocol Communications Controller (MPCC) is a monolithic n-channel MOS LSI circuit that formats, transmits and receives synchronous serial data while supporting bit-oriented or byte control protocols. The chip is TTL compatible, operates from a single +5V supply, and can interface to a processor with an 8 or 16-bit bidirectional data bus.

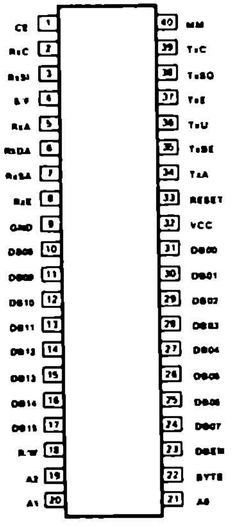
FEATURES

- DC to 1Mbps data rate, 2652-1 to 2Mbps
- Bit-oriented protocols (BOP): SDLC, ADCCP, HDLC
- Byte-control protocols (BCP): DDCMP, BISYNC (external CRC)
- Programmable operation
  - 8 or 16-bit tri-state data bus
  - Error control—CRC or VRC or none
  - Character length—1 to 8 bits for BOP or 5 to 8 bits for BCP
  - SYNC or secondary station address comparison for BCP-BOP
  - Idle transmission of SYNC/FLAG or MARK for BCP-BOP
- Automatic detection and generation of special BOP control sequences, i.e., FLAG, ABORT, GA
- Zero insertion and deletion for BOP
- Short character detection for last BOP data character
- SYNC generation, detection, and stripping for BCP
- Maintenance Mode for self-testing
- TTL compatible
- Single +5V supply

APPLICATIONS

- Intelligent terminals
- Line controllers
- Network processors
- Front end communications
- Remote data concentrators
- Communication test equipment
- Computer to computer links

PIN CONFIGURATION


CE 1
RxC 2
RxSI 3
S/F 4
RxA 5
RxDA 6
RxSA 7
RxE 8
GND 9
DB08 10
DB09 11
DB10 12
DB11 13
DB12 14
DB13 15
DB14 16
DB15 17
R/W 18
A2 19
A1 20
40 MM
39 TxC
38 TxSO
37 TxE
36 TxU
35 TxBE
34 TxA
33 RESET
32 VCC
31 DB00
30 DB01
29 DB02
28 DB03
27 DB04
26 DB05
25 DB06
24 DB07
23 DBEN
22 BYTE
21 A0

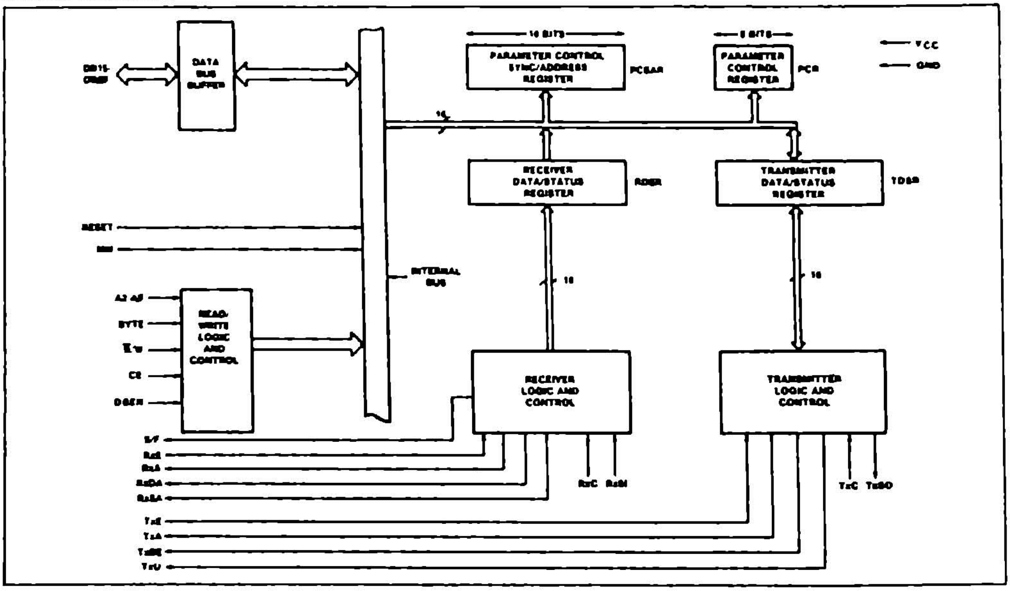
TOP VIEW

NOTE
00 is least significant bit, highest number (that is DB15, A2) is most significant bit

ORDERING CODE

PACKAGE	V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
	1.0MHz Clock Max	2.0MHz Clock Max
Ceramic DIP	SC2652C1I40	SC2652C2I40
Plastic DIP	SC2652C1N40	SC2652C2N40

BLOCK DIAGRAM


DB15-DB00
DATA BUS BUFFER
16 BITS
PARAMETER CONTROL SYNC/ADDRESS REGISTER
PCSAR
8 BITS
PARAMETER CONTROL REGISTER
PCR
VCC
GND
16
RECEIVER DATA/STATUS REGISTER
RDSR
TRANSMITTER DATA/STATUS REGISTER
TDSR
RESET
MM
INTERNAL BUS
16
16
A2-A0
BYTE
R/W
CE
DBEN
READ/WRITE LOGIC AND CONTROL
RECEIVER LOGIC AND CONTROL
TRANSMITTER LOGIC AND CONTROL
S/F
RxE
RxA
RxDA
RxSA
TxE
TxA
TxBE
TxU
RxC
RxSI
TxC
TxSO

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER (MPCC) SC2652

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DB15-DB00	17-10 24-31	I/O	**Data Bus:** DB07-DB00 contain bidirectional data while DB15-DB08 contain control and status information to or from the processor Corresponding bits of the high and low order bytes can be WIRE OR'ed onto an 8-bit bus The data bus is floating if either CE or DBEN are low.
A2-A0	19-21	I	**Address Bus:** A2-A0 select internal registers The four 16-bit registers can be addressed on a word or byte basis See Register Address section
BYTE	22	I	**Byte:** Single byte (8 bit) data bus transfers are specified when this input is high. A low level specifies 16 bit data bus transfers
CE	1	I	**Chip Enable:** A high input permits a data bus operation when DBEN is activated
$\overline{R}$/W	18	I	**Read/Write:** $\overline{R}$/W controls the direction of data bus transfer When high, the data is to be loaded into the addressed register A low input causes the contents of the addressed register to be presented on the data bus
DBEN	23	I	**Data Bus Enable:** After A2-A0, CE, BYTE and $\overline{R}$/W are set up, DBEN may be strobed. During a read, the 3-state data bus (DB) is enabled with information for the processor. During a write, the stable data is loaded into the addressed register and TxBE will be reset if TDSR was addressed.
RESET	33	I	**Reset:** A high level initializes all internal registers (to zero) and timing.
MM	40	I	**Maintenance Mode:** MM internally gates TxSO back to RxSI and $\overline{TxC}$ to RxC for off line diagnostic purposes. The RxC and RxSI inputs are disabled and TxSO is high when MM is asserted.
RxE	8	I	**Receiver Enable:** A high level input permits the processing of RxSI data. A low level disables the receiver logic and initializes all receiver registers and timing.
RxA	5	O	**Receiver Active:** RxA is asserted when the first data character of a message is ready for the processor. In the BOP mode this character is the address The received address must match the secondary station address if the MPCC is a secondary station. In BCP mode, if strip-SYNC ($PCSAR_{13}$) is set, the first non-SYNC character is the first data character; if strip-SYNC is zero, the character following the second SYNC is the first data character. In the BOP mode, the closing FLAG resets RxA. In the BCP mode, RxA is reset by a low level at RxE.
RxDA*	6	O	**Receiver Data Available:** RxDA is asserted when an assembled character is in $RDSR_L$ and is ready to be presented to the processor This output is reset when $RDSR_L$ is read.
RxC	2	I	**Receiver Clock:** RxC(1X) provides timing for the receiver logic. The positive going edge shifts serial data into the RxSR from RxSI.
S/F	4	O	**SYNC/FLAG:** S/F is asserted for one RxC clock time when a SYNC or FLAG character is detected
RxSA*	7	O	**Receiver Status Available:** RxSA is asserted when there is a zero to one transition of any bit in $RDSR_H$ except for RSOM. It is cleared when $RDSR_H$ is read
RxSI	3	I	**Receiver Serial Input:** RxSI is the received serial data. Mark = '1', space = '0'.
TxE	37	I	**Transmitter Enable:** A high level input enables the transmitter data path between $TDSR_L$ and TxSO At the end of a message, a low level input causes TxSO = 1 (mark) and TxA = 0 after the closing FLAG (BOP) or last character (BCP) is output on TxSO.
TxA	34	O	**Transmitter Active:** TxA is asserted after TSOM ($TDSR_8$) is set and TxE is raised. This output will reset when TxE is low and the closing FLAG (BOP) or last character (BCP) has been output on TxSO.
TxBE*	35	O	**Transmitter Buffer Empty:** TxBE is asserted when the TDSR is ready to be loaded with new control information or data The processor should respond by loading the TDSR which resets TxBE
TxU*	36	O	**Transmitter Underrun:** TxU is asserted during a transmit sequence when the service of TxBE has been delayed for one character time This indicates the processor is not keeping up with the transmitter. Line fill depends on $PCSAR_{11}$. TxU is reset by RESET or setting of TSOM ($TDSR_8$), synchronized by the falling edge of TxC
TxC	39	I	**Transmitter Clock:** TxC (1X) provides timing for the transmitter logic. The positive going edge shifts data out of the TxSR to TxSO
TxSO	38	O	**Transmitter Serial Output.** TxSO is the transmitted serial data Mark = '1', space = '0'.
V_{CC}	32	I	**+5V:** Power supply
GND	9	I	**Ground:** 0V reference ground

*Indicates possible interrupt signal

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER (MPCC) SC2652

REGISTERS		NO. OF BITS	DESCRIPTION*
Addressable			
PCSAR	Parameter Control Sync/Address Register	16	PCSAR_H and PCR contain parameters common to the receiver and transmitter. PCSAR_L contains a programmable SYNC character (BCP) or secondary station address (BOP)
PCR	Parameter Control Register	8	
RDSR	Receive Data/Status Register	16	RDSR_H contains receiver status information. RDSR_L = RxDB contains the received assembled character.
TDSR	Transmit Data/Status Register	16	TDSR_H contains transmitter command and status information. TDSRL = TxDB contains the character to be transmitted
Internal			
CCSR	Control Character Shift Register	8	These registers are used for character assembly (CCSR, HSR, RxSR), disassembly (TxSR), and CRC accumulation/generation (RxCRC, TxCRC).
HSR	Holding Shift Register	16	
RxSR	Receiver Shift Register	8	
TxSR	Transmitter Shift Register	8	
RxCRC	Receiver CRC Accumulation Register	16	
TxCRC	Transmitter CRC Generation Register	16	

NOTE

*H = High byte - bits 15-8
L = Low byte - bits 7-0

Table 1 GLOSSARY

CHARACTER	DESCRIPTION
FCS	Frame Check Sequence is transmitted/received as 16 bits following the last data character of a BOP message. The divisor is usually CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) with dividend preset to 1's but can be otherwise determined by ECM. The inverted remainder is transmitted as the FCS.
BCC	Block Check Character is transmitted/received as two successive characters following the last data character of a BCP message. The polynomial is CRC-16 ($X^{16} + X^{15} + X^2 + 1$) or CRC-CCITT with dividend preset to 0's (as specified by ECM). The true remainder is transmitted as the BCC.

Table 2 ERROR CONTROL

FUNCTIONAL DESCRIPTION

The MPCC can be functionally partitioned into receiver logic, transmitter logic, registers that can be read or loaded by the processor, and data bus control circuitry. The register bit formats are shown in figure 1 while the receiver and transmitter data paths are depicted in figures 2 and 3.

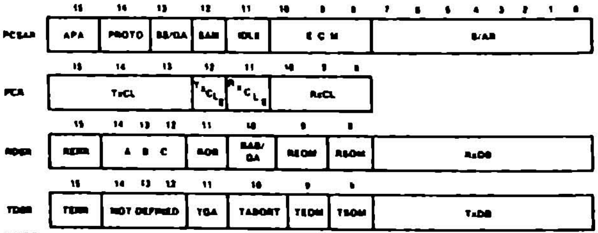
OPERATION	BIT PATTERN	FUNCTION
BOP		
FLAG	01111110	Frame message
ABORT	11111111 generation 01111111 detection	Terminate communication
GA	01111111	Terminate loop mode repeater function
Address	(PCSAR_L)[1]	Secondary station address
BCP		
SYNC	(PCSAR_L) or (TxDB)[2] generation	Character synchronization

NOTES

1 (α) refers to contents of α
2 For IDLE = 0 or 1 respectively

Table 3 SPECIAL CHARACTERS

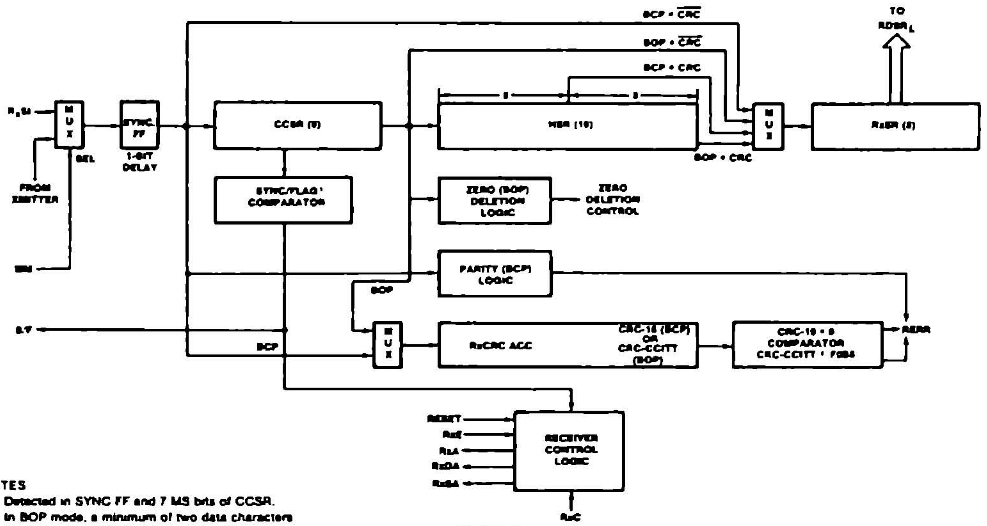
SHORT FORM REGISTER BIT FORMATS



NOTE
Refer to Register Formats for mneumonics and description

Figure 1

MPCC RECEIVER DATA PATH

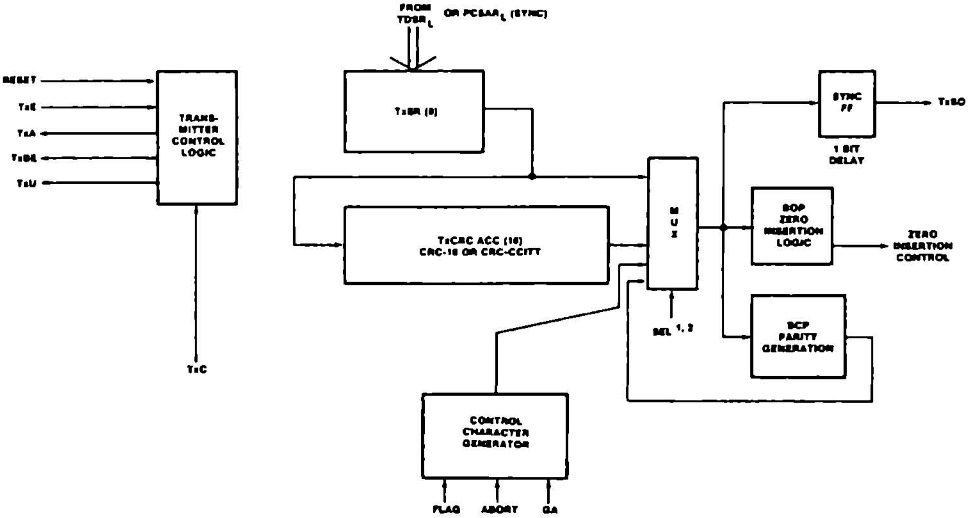

BCP • $\overline{CRC}$
BOP • $\overline{CRC}$
BCP • CRC
BOP • CRC
TO RDSR$_L$
R_xSI
MUX
SEL
FROM XMITTER
MM
SYNC FF
1-BIT DELAY
CCSR (8)
SYNC/FLAG¹ COMPARATOR
HSR (16)
8
8
MUX
RxSR (8)
ZERO (BOP) DELETION LOGIC
ZERO DELETION CONTROL
PARITY (BCP) LOGIC
BOP
S/F
BCP
MUX
RxCRC ACC
CRC-16 (BCP) OR CRC-CCITT (BOP)
CRC-16 = 0 COMPARATOR CRC-CCITT = F0B8
RERR
RESET
RxE
RxA
RxDA
RxSA
RECEIVER CONTROL LOGIC
RxC

NOTES

1. Detected in SYNC FF and 7 MS bits of CCSR.
2. In BOP mode, a minimum of two data characters must be received to turn the receiver active

Figure 2

MPCC TRANSMITTER DATA PATH


FROM TDSR$_L$ OR PCSAR$_L$ (SYNC)
RESET
TxE
TxA
TxBE
TxU
TRANSMITTER CONTROL LOGIC
TxC
TxSR (8)
TxCRC ACC (16) CRC-16 OR CRC-CCITT
MUX
SEL 1, 2
SYNC FF
1 BIT DELAY
TxSO
BOP ZERO INSERTION LOGIC
ZERO INSERTION CONTROL
BCP PARITY GENERATION
CONTROL CHARACTER GENERATOR
FLAG
ABORT
GA

NOTES

1. TxCRC selected if TEOM = 1 and the last data character has been shifted out of TxSR.
2. In BCP, parity selected will be generated after each character is shifted out of TxSR

Figure 3

RECEIVER OPERATION

General

After initializing the parameter control registers (PCSAR and PCR), the RxE input must be set high to enable the receiver data path. The serial data on the RxSI is synchronized and shifted into an 8-bit Control Character Shift Register (CCSR) on the rising edge of RxC. A comparison between CCSR contents and the FLAG (BOP) or SYNC (BCP) character is made until a match is found. At that time, the S/F output is asserted for one RxC time and the 16-bit Holding Shift Register (HSR) is enabled. The receiver then operates as described below.

BOP Operation

A flow chart of receiver operation in BOP mode appears in figure 4. Zero deletion (after five ones are received) is implemented on the received serial data so that a data character will not be interpreted as a FLAG, ABORT, or GA. Bits following the FLAG are shifted through the CCSR, HSR, and into the Receiver Shift Register (RxSR). A character will be assembled in the RxSR and transferred to the $RDSR_L$ for presentation to the processor. At that time the RxDA output will be asserted and the processor must take the character no later than one RxC time after the next character is assembled in the RxSR. If not, an overrun ($RDSR_{11}$ = 1) will occur and succeeding characters will be lost.

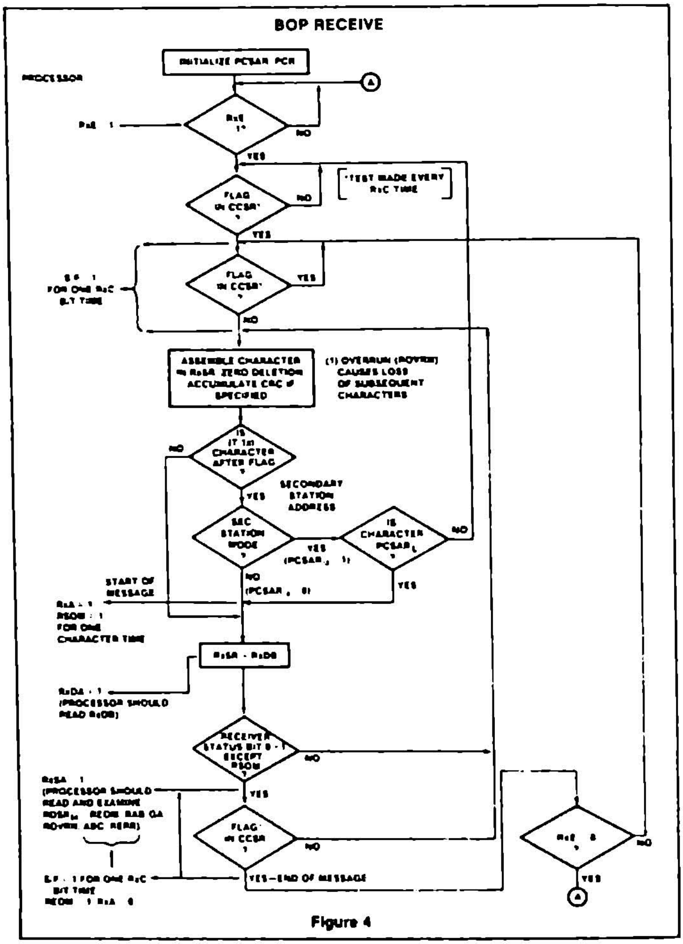

BOP RECEIVE
PROCESSOR
INITIALIZE PCSAR PCR
A
RxE 1
RxE 1?
NO
YES
*TEST MADE EVERY RxC TIME
FLAG IN CCSR?
NO
YES
S/F 1 FOR ONE RxC BIT TIME
FLAG IN CCSR?
YES
NO
ASSEMBLE CHARACTER IN RxSR ZERO DELETION ACCUMULATE CRC IF SPECIFIED
(1) OVERRUN (ROVRN) CAUSES LOSS OF SUBSEQUENT CHARACTERS
IS IT 1st CHARACTER AFTER FLAG?
NO
YES
SECONDARY STATION ADDRESS
SEC STATION MODE?
YES (PCSAR12 1)
IS CHARACTER PCSARL?
NO
YES
NO (PCSAR12 0)
START OF MESSAGE
RxA 1
RSOM 1
FOR ONE CHARACTER TIME
RxSR RxDB
RxDA 1 (PROCESSOR SHOULD READ RxDB)
RECEIVER STATUS BIT 8-15 EXCEPT RSOM?
NO
YES
RxSA 1 (PROCESSOR SHOULD READ AND EXAMINE RDSRH REOM RAB GA ROVRN ABC RERR)
FLAG IN CCSR?
NO
YES—END OF MESSAGE
S/F 1 FOR ONE RxC BIT TIME
REOM 1 RxA 0
RxE 0?
NO
YES
A

Figure 4

The first character following the FLAG is the secondary station address. If the MPCC is a secondary station ($PCSAR_{12}$ = 1), the contents of RxSR are compared with the address stored in $PCSAR_L$. A match indicates the forthcoming message is intended for the station, the RxA output is asserted, the character is loaded into $RDSR_L$. RxDA is asserted and the Receive Start of Message bit (RSOM) is set. No match indicates that another station is being addressed and the receiver searches for the next FLAG.

If the MPCC is a primary station ($PCSAR_{12}$ = 0), no secondary address check is made; RxA is asserted and RSOM is set once the first non-FLAG character has been loaded into $RDSR_L$ and RxDA has been asserted. Extended address field can be supported by software if $PCSAR_{12}$ = 0.

When the 8 bits following the address character have been loaded into $RDSR_L$ and RxDA has been asserted, RSOM will be cleared. The processor should read this 8-bit character and interpret it as the Control field.

Received serial data that follows is read and interpreted as the Information field by the processor. It will be assembled into character lengths as specified by PCR_{8-10}. As before, RxDA is asserted each time a character has been transferred into $RDSR_L$ and is cleared when $RDSR_L$ is read by the processor. $RDSR_H$ should only be read when RxSA is asserted. This occurs on a zero to one transition of any bit in $RDSR_H$ except for RSOM. RxSA and all bits in $RDSR_H$ except RSOM are cleared when $RDSR_H$ is read. The processor should check $RDSR_{9-15}$ each time RxSA is asserted. If $RDSR_9$ is set, then $RDSR_{12-15}$ should be examined.

Receiver character length may be changed dynamically in response to RxDA: read the character in RxDB and write the new character length into RxCL. The character length will be changed on the next receiver character boundary. A received residual (short) character will be transferred into RxDB after the previous character in RxDB has been read, i.e. there will not be an overrun. In general the last two characters are protected from overrun.

The CRC-CCITT, if specified by $PCSAR_{8-10}$, is accumulated in RxCRC on each character following the FLAG. When the closing FLAG is detected in the CCSR, the received CRC is in the 16-bit HSR. At that time, the Receive End of Message bit (REOM) will be set; RxSA and RxDA will be asserted. The processor should read the last data character in $RDSR_L$ and the receiver status in $RDSR_{8-15}$. If $RDSR_{15}$ = 1, there has been a transmission error, the accumulated CRC-CCITT is incorrect. If $RDSR_{12-14}$ ≠ 0, the last data character is not of prescribed length. Neither the received CRC nor closing FLAG are presented to the processor. The processor may drop RxE or leave it active at the end of the received message.

BCP Operation

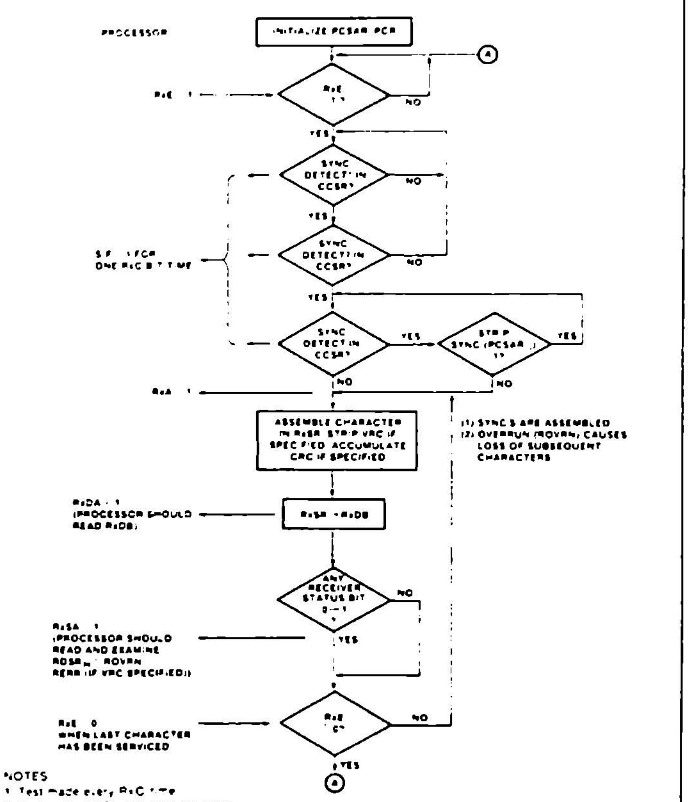
The operation of the receiver in BCP mode is shown in figure 5. The receiver initially searches for two successive SYNC characters, of length specified by PCR_{8-10}, that match the contents of PCSARL. The next non-SYNC character or next SYNC character, if stripping is not specified ($PCSAR_{13}$ = 0), causes RxA to be asserted and enables the receiver data path. Once enabled, all characters are assembled in RxSR and loaded into $RDSR_L$. RxDA is active when a character is available in $RDSR_L$. RxSA is active on a 0 to 1 transition of any bit in $RDSR_H$. The signals are cleared when $RDSR_L$ or $RDSR_H$ are read respectively.

If CRC-16 error control is specified by $PCSAR_{8-10}$, the processor must determine the last character received prior to the CRC field. When that character is loaded into $RDSR_L$ and RxDA is asserted, the received CRC will be in CCSR and HSR_L. To check for a transmission error, the processor must read the receiver status ($RDSR_H$) and examine $RDSR_{15}$. This bit will be set for one character time if an error free message has been received. If $RDSR_{15}$ = 0, the CRC-16 is in error. The state of RDSR15 in BCP CRC mode does not set RxSA. Note that this bit should be examined only at the end of a message. The accumulated CRC will include all characters starting with the first non-SYNC character if $PCSAR_{13}$ = 1, or the character after the opening two SYNC's if $PCSAR_{13}$ = 0. This necessitates external CRC generation/checking when supporting IBM's BISYNC. This can be accomplished using the Signetics 2653 Polynomial Generator/Checker. See Typical Applications.

If VRC had been selected for error control, parity (odd or even) is regenerated on each character and checked when the parity bit is received. A discrepancy causes $RDSR_{15}$ to be set and RxSA to be asserted. This must be sensed by the processor. The received parity bit is stripped before the character is presented to the processor.

When the processor has read the last character of the message, it should drop RxE which disables the receiver logic and initializes all receiver registers and timing.

BCP RECEIVE


PROCESSOR
INITIALIZE PCSAR PCR
A
RxE = 1
RxE = 1?
NO
YES
SYNC DETECT¹ IN CCSR?
NO
YES
SF = 1 FOR ONE RxC BIT TIME
SYNC DETECT¹ IN CCSR?
NO
YES
SYNC DETECT² IN CCSR?
YES
STRIP SYNC (PCSAR = 1)?
YES
NO
NO
RxA = 1
ASSEMBLE CHARACTER IN RxSR STRIP VRC IF SPECIFIED ACCUMULATE CRC IF SPECIFIED
(1) SYNC'S ARE ASSEMBLED
(2) OVERRUN (ROVRN) CAUSES LOSS OF SUBSEQUENT CHARACTERS
RxDA = 1 (PROCESSOR SHOULD READ RxDB)
RxSR → RxDB
ANY RECEIVER STATUS BIT 0→1?
NO
RxSA = 1 (PROCESSOR SHOULD READ AND EXAMINE $RDSR_H$ ROVRN RERR IF VRC SPECIFIED))
YES
RxE = 0 WHEN LAST CHARACTER HAS BEEN SERVICED
RxE = 0?
NO
YES
A

NOTES
1 Test made every RxC time
2 Test made on Rx character boundary

Figure 5

TRANSMITTER OPERATION

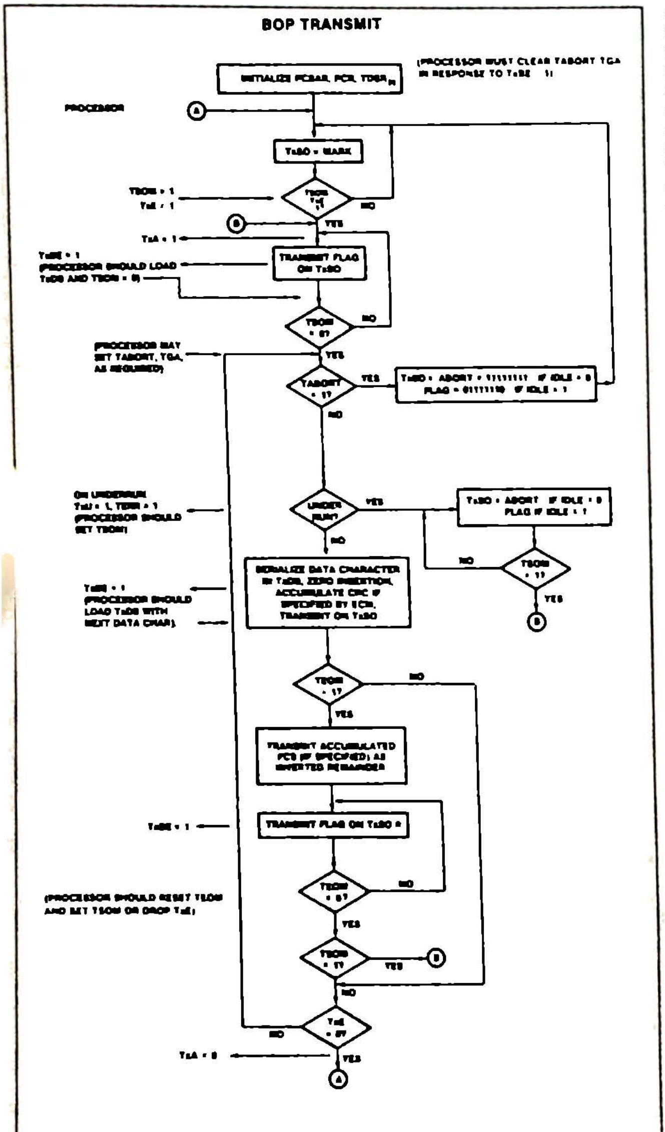
General

After the parameter control registers (PCSAR and PCR) have been initialized TxSO is held at mark until TSOM ($TDSR_8$) is set and TxE is raised. Then, transmitter operation depends on protocol mode.

BOP Operation

Transmitter operation for BOP is shown in figure 6. A FLAG is sent after the processor sets the Transmit Start of Message bit (TSOM) and raises TxE. The FLAG is used to synchronize the message that follows. TxA will also be asserted. When TxBE is asserted by the MPCC, the processor should load $TDSR_L$ with the first character of the message. TSOM should be cleared at the same time $TDSR_L$ is loaded (16-bit data bus) or immediately thereafter (8-bit data bus). FLAGs are sent as long as TSOM = 1. For counting the number of FLAGs the processor should reassert TSOM in response to the assertion of TxBE.

All succeeding characters are loaded into $TDSR_L$ by the processor when TxBE = 1. Each character is serialized in TxSR and transmitted on TxSO. Internal zero insertion logic stuffs a "0" into the serial bit stream after five successive "1s" are sent. This insures a data character will not match a FLAG, ABORT, or GA reserved control character. As each character is transmitted, the Frame Check Sequence (FCS) is gener-


BOP TRANSMIT
INITIALIZE PCSAR, PCR, TDSR$_H$
(PROCESSOR MUST CLEAR TABORT TGA IN RESPONSE TO TxBE = 1)
PROCESSOR
A
TxSO = MARK
TSOM = 1
TxE = 1
TSOM TxE (1)
NO
B
YES
TxA = 1
TxBE = 1
(PROCESSOR SHOULD LOAD TxDB AND TSOM = 0)
TRANSMIT FLAG ON TxSO
TSOM = 0?
NO
(PROCESSOR MAY SET TABORT, TGA, AS REQUIRED)
YES
TABORT = 1?
YES
TxSO = ABORT = 11111111 IF IDLE = 0
FLAG = 01111110 IF IDLE = 1
NO
ON UNDERRUN
TxU = 1, TERR = 1
(PROCESSOR SHOULD SET TSOM)
UNDER RUN?
YES
TxSO = ABORT IF IDLE = 0
FLAG IF IDLE = 1
NO
TSOM = 1?
YES
B
NO
TxBE = 1
(PROCESSOR SHOULD LOAD TxDB WITH NEXT DATA CHAR)
SERIALIZE DATA CHARACTER IN TxDB, ZERO INSERTION, ACCUMULATE CRC IF SPECIFIED BY ECM, TRANSMIT ON TxSO
TEOM = 1?
NO
YES
TRANSMIT ACCUMULATED FCS (IF SPECIFIED) AS INVERTED REMAINDER
TxBE = 1
TRANSMIT FLAG ON TxSO *
(PROCESSOR SHOULD RESET TEOM AND SET TSOM OR DROP TxE)
TEOM = 0?
NO
YES
TSOM = 1?
YES
B
NO
TxE = 0?
NO
YES
TxA = 0
A

* GA will be transmitted if TGA is set together with TEOM

Figure 6

ated as specified by Error Control Mode (PCSAR$_{8-10}$). The FCS should be the CRC-CCITT polynomial ($X^{16} + X^{12} + X^5 + 1$) preset to 1s. If an underrun occurs (processor is not keeping up with the transmitter), TxU and TERR (TDSR$_{15}$) will be asserted with ABORT or FLAG used as the TxSO line fill depending on the state of IDLE (PCSAR$_{11}$). The processor must set TSOM to reset the underrun condition. To retransmit the message, the processor should proceed with the normal start of message sequence.

A residual character of 1 to 7 bits may be transmitted at the end of the Information field. In response to TxBE, write the residual character length into TxCL and load TxDB with the residual character. Dynamic alteration of character length should be done in exactly the same sequence. The character length will be changed on the next transmit character boundary.

After the last data character has been loaded into TDSR$_L$ and sent to TxSR (TxBE = 1), the processor should set TEOM (TDSR$_9$). The MPCC will finish transmitting the last character followed by the FCS and the closing FLAG. The processor should clear TEOM and drop TxE when the next TxBE is asserted. This corresponds to the start of closing FLAG transmission. When TxE has been dropped, TxA will be low $1\frac{1}{2}$ bit times after the last bit of the closing FLAG has been transmitted. TxSO will be marked after the closing FLAG has been transmitted.

If TxE and TEOM are high, the transmitter continues to send FLAGs. The processor may initiate the next message by resetting TEOM and setting TSOM, or by loading TDSR$_L$ with a data character and then simply resetting TEOM (without setting TSOM).

BCP Operation

Transmitter operation for BCP mode is shown in figure 7. TxA will be asserted after TSOM = 1 and TxE is raised. At that time SYNC characters are sent from PCSAR$_L$ or TDSR$_L$ (IDLE = 0 or 1) as long as TSOM = 1. TxBE is asserted at the start of transmission of the first SYNC character. For counting the number of SYNC's, the processor should reassert TSOM in response to the assertion of TxBE. When TSOM = 0 transmission is from TDSR$_L$, which must be loaded with characters from the processor each time TxBE is asserted. If this loading is delayed for more than one character time, an underrun results: TxU and TERR are asserted and the TxSO line fill depend on IDLE (PCSAR$_{11}$). The processor must set TSOM

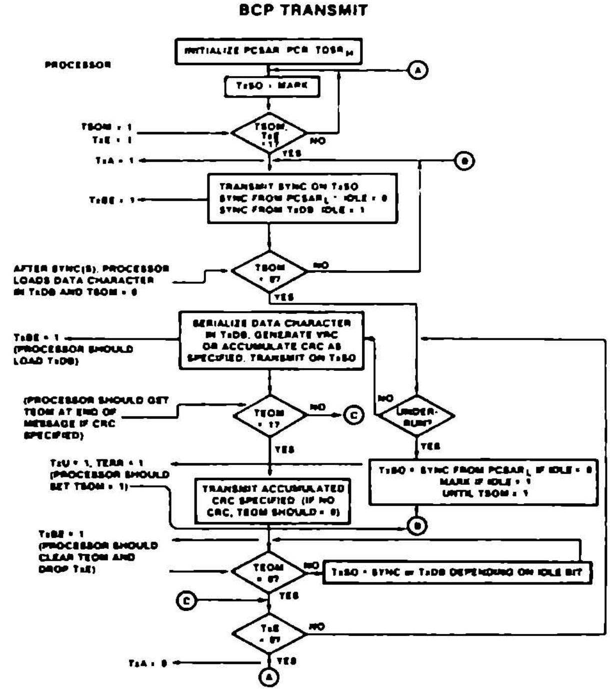

BCP TRANSMIT
PROCESSOR
INITIALIZE PCSAR PCR TDSR$_H$
TxSO = MARK
TSOM = 1
TxE = 1
TSOM, TxE = 1?
NO
YES
TxA = 1
TRANSMIT SYNC ON TxSO
SYNC FROM PCSAR$_L$ IF IDLE = 0
SYNC FROM TxDB IDLE = 1
TxBE = 1
AFTER SYNC(S), PROCESSOR LOADS DATA CHARACTER IN TxDB AND TSOM = 0
TSOM = 0?
SERIALIZE DATA CHARACTER IN TxDB, GENERATE VRC OR ACCUMULATE CRC AS SPECIFIED. TRANSMIT ON TxSO
TxBE = 1 (PROCESSOR SHOULD LOAD TxDB)
(PROCESSOR SHOULD SET TEOM AT END OF MESSAGE IF CRC SPECIFIED)
TEOM = 1?
UNDER-RUN?
TxU = 1, TERR = 1 (PROCESSOR SHOULD SET TSOM = 1)
TxSO = SYNC FROM PCSAR$_L$ IF IDLE = 0
MARK IF IDLE = 1
UNTIL TSOM = 1
TRANSMIT ACCUMULATED CRC SPECIFIED (IF NO CRC, TEOM SHOULD = 0)
TxBE = 1 (PROCESSOR SHOULD CLEAR TEOM AND DROP TxE)
TEOM = 0?
TxSO = SYNC or TxDB DEPENDING ON IDLE BIT
TxE = 0?
TxA = 0
A
B
C

Figure 7

and retransmit the message to recover. This is not compatible with IBM's BISYNC, so that the user must not underrun when supporting that protocol.

CRC-16, if specified by PCSAR$_{8-10}$, is generated on each character transmitted from TDSR$_L$ when TSOM = 0. The processor must set TEOM = 1 after the last data character has been sent to TxSR (TxBE = 1). The MPCC will finish transmitting the last data character and the CRC-16 field before sending SYNC characters which are transmitted as long as TEOM = 1. If SYNCs are not desired after CRC-16 transmission, the processor should clear TEOM and lower TxE when the TxBE corresponding to the start of CRC-16 transmission is asserted. When TEOM = 0, the line is marked and a new message may be inlated by setting TSOM and raising TxE.

If VRC is specified, it is generated on each data character and the data character length must not exceed 7 bits. For software LRC or CRC, TEOM should be set only if SYNC's are required at the end of the message block.

Special Case

The capability to transmit 16 spaces is provided for line turnaround in half duplex mode or for a control recovery situation. This is achieved by setting TSOM and TEOM, clearing TEOM when TxBE = 1, and proceeding as required.

PROGRAMMING

Prior to initiating data transmission or reception, PCSAR and PCR must be loaded with control information from the processor. The contents of these registers (see Register Format section) will configure the MPCC for the user's specific data communication environment. These registers should be loaded during power-on initialization and after a reset operation. They can be changed at any time that the respective transmitter or receiver is disabled.

The default value for all registers is zero. This corresponds to BOP, primary station mode, 8-bit character length, FCS = CRC-CCITT preset to 1s.

For BOP mode the character length register (PCR) may be set to the desired values during system initialization. The address and control fields will automatically be 8-bits. If a residual character is to be transmitted, TxCL should be changed to the residual character length prior to transmission of that character.

DATA BUS CONTROL

The processor must set up the MPCC register address (A2-A0), chip enable (CE), byte select (BYTE), and read/write (R̄/W) inputs before each data bus transfer operation.

During a read operation (R̄/W = 0), the leading edge of DBEN will initiate an MPCC read cycle. The addressed register will place its contents on the data bus. If BYTE = 1, the 8-bit byte is placed on DB15–08 or DB07-00 depending on the H/L status of the register addressed. Unused bits in RDSR$_L$ are zero. If BYTE = 0, all 16 bits (DB15-00) contain MPCC information. The trailing edge of DBEN will reset RxDA and/or RxSA if RDSR$_L$ or RDSR$_H$ is addressed respectively.

DBEN acts as the enable and strobe so that the MPCC will not begin its internal read cycle until DBEN is asserted.

During a write operation (R̄/W = 1), data must be stable on DB$_{15-08}$ and/or DB$_{07-00}$ prior to the leading edge of DBEN. The stable data is strobed into the addressed register by DBEN. TxBE will be cleared if the addressed register was TDSR$_H$ or TDSR$_L$.

	A2	A1	A0	REGISTER
BYTE = 0	16-BIT DATA BUS = DB_{15} - DB_{00}			
	0	0	X	RDSR
	0	1	X	TDSR
	1	0	X	PCSAR
	1	1	X	PCR*
BYTE = 1	8-BIT DATA BUS = $DB_{7\text{-}0}$ or $DB_{15\text{-}8}$**			
	0	0	0	RDSR$_L$
	0	0	1	RDSR$_H$
	0	1	0	TDSR$_L$
	0	1	1	TDSR$_H$
	1	0	0	PCSAR$_L$
	1	0	1	PCSAR$_H$
	1	1	0	PCR$_L$*
	1	1	1	PCR$_H$

NOTES

* PCR lower byte does not exist. It will be all '0's when read

** Corresponding high and low order pins must be tied together

Table 4 MPCC REGISTER ADDRESSING

BIT	NAME	MODE	FUNCTION
00-07	Not Defined		
08-10	RxCL	BOP/BCP	Receiver Character Length is loaded by the processor when RxCLE = 0. The character length is valid after transmission of single byte address and control fields have been received. **10 9 8 Char. length (bits)** 0 0 0 8 0 0 1 1 0 1 0 2 0 1 1 3 1 0 0 4 1 0 1 5 1 1 0 6 1 1 1 7
11	RxCLE	BOP/BCP	Receiver Character Length Enable should be zero when the processor loads RxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
12	TxCLE	BOP/BCP	Transmitter Character Length Enable should be zero when the processor loads TxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
13-15	TxCL	BOP/BCP	Transmitter Character Length is loaded by the processor when TxCLE = 0. Character bit length specification format is identical to RxCL. It is valid after transmission of single byte address and control fields.

Table 5 PARAMETER CONTROL REGISTER (PCR)-(R/W)

<table>
<tr><th>BIT</th><th>NAME</th><th>MODE</th><th>FUNCTION</th></tr>
<tr><td>00-07</td><td>S/AR</td><td>BOP
BCP</td><td>SYNC/ADDRESS Register. Contains the secondary station address if the MPCC is a secondary station. The contents of this register is compared with the first received non-FLAG character to determine if the message is meant for this station.
SYNC character is loaded into this register by the processor. It is used for receive and transmit bit synchronization with bit length specified by RxCL and TxCL.</td></tr>
<tr><td>08-10</td><td>ECM</td><td>BOP/BCP</td><td><table>
<tr><th>Error Control Mode</th><th>10</th><th>9</th><th>8</th><th>Suggested Mode</th><th>Char. length</th></tr>
<tr><td>CRC-CCITT preset to 1's</td><td>0</td><td>0</td><td>0</td><td>BOP</td><td>1-8</td></tr>
<tr><td>CRC-CCITT preset to 0's</td><td>0</td><td>0</td><td>1</td><td>BCP</td><td>8</td></tr>
<tr><td>Not used</td><td>0</td><td>1</td><td>0</td><td>---</td><td></td></tr>
<tr><td>CRC-16 preset to 0's</td><td>0</td><td>1</td><td>1</td><td>BCP</td><td>8</td></tr>
<tr><td>VRC odd</td><td>1</td><td>0</td><td>0</td><td>BCP</td><td>5-7</td></tr>
<tr><td>VRC even</td><td>1</td><td>0</td><td>1</td><td>BCP</td><td>5-7</td></tr>
<tr><td>Not used</td><td>1</td><td>1</td><td>0</td><td>---</td><td></td></tr>
<tr><td>No error control</td><td>1</td><td>1</td><td>1</td><td>BCP/BOP</td><td>5-8</td></tr>
</table>ECM should be loaded by the processor during initialization or when both data paths are idle.</td></tr>
<tr><td>11</td><td>IDLE</td><td>BOP
BCP</td><td>Determines line fill character to be used if transmitter underrun occurs (TxU asserted and TERR set) and transmission of special characters for BOP/BCP.
IDLE = 0, transmit ABORT characters during underrun and when TABORT = 1
IDLE = 1, transmit FLAG characters during underrun and when TABORT = 1
IDLE = 0 transmit initial SYNC characters and underrun line fill characters from the S/AR
IDLE = 1 transmit initial SYNC characters from TxDB and marks TxSO during underrun</td></tr>
<tr><td>12</td><td>SAM</td><td>BOP</td><td>Secondary Address Mode = 1 if the MPCC is a secondary station. This facilitates automatic recognition of the received secondary station address. When transmitting, the processor must load the secondary address into TxDB.
SAM = 0 inhibits the received secondary address comparison which serves to activate the receiver after the first non-FLAG character has been received.</td></tr>
<tr><td>13</td><td>SS/GA</td><td>BOP
BCP</td><td>Strip SYNC/Go Ahead. Operation depends on mode.
SS/GA = 1 is used for loop mode only and enables GA detection. When a GA is detected as a closing character, REOM and RAB/GA will be set and the processor should terminate the repeater function. SS/GA = 0 is the normal mode which enables ABORT detection. It causes the receiver to terminate the frame upon detection of an ABORT or FLAG.
SS/GA = 1, causes the receiver to strip SYNC's immediately following the first two SYNC's detected. SYNC's in the middle of a message will not be stripped. SS/GA = 0, presents any SYNC's after the initial two SYNC's to the processor.</td></tr>
<tr><td>14</td><td>PROTO</td><td>BOP
BCP</td><td>Determines MPCC Protocol mode
PROTO = 0
PROTO = 1</td></tr>
<tr><td>15</td><td>APA</td><td>BOP</td><td>All Parties Address. If this bit is set, the receiver data path is enabled by an address field of '11111111' as well as the normal secondary station address.</td></tr>
</table>

Table 6 PARAMETER CONTROL SYNC/ADDRESS REGISTER (PCSAR)-(R/W)

BIT	NAME	MODE	FUNCTION
00-07	TxDB	BOP/BCP	Transmit Data Buffer. Contains processor loaded characters to be serialized in TxSR and transmitted on TxSO.
08	TSOM	BOP BCP	Transmitter Start of Message. Set by the processor to initiate message transmission provided TxE = 1. TSOM = 1 generates FLAGs. When TSOM = 0 transmission is from TxDB and FCS generation (if specified) begins. FCS, as specified by $PCSAR_{8-10}$, should be CRC-CCITT preset to 1's. TSOM = 1 generates SYNCs from $PCSAR_L$ or transmits from TxDB for IDLE = 0 or 1 respectively. When TSOM = 0 transmission is from TxDB and CRC generation (if specified) begins.

Table 7 TRANSMIT DATA/STATUS REGISTER (TDSR) (R/W except TDSR 15)

BIT	NAME	MODE	FUNCTION
09	TEOM		Transmit End of Message. Used to terminate a transmitted message.
		BOP	TEOM = 1 causes the FCS and the closing FLAG to be transmitted following the transmission of the data character in TxSR. FLAGs are transmitted until TEOM = 0. ABORT or GA are transmitted if TABORT or TGA are set when TEOM = 1
		BCP	TEOM = 1 causes CRC-16 to be transmitted (if selected) followed by SYNCs from $PCSAR_L$ or TxDB (IDLE = 0 or 1) Clearing TEOM prior to the end of CRC-16 transmission (when TxBE = 1) causes TxSO to be marked following the CRC-16. TxE must be dropped before a new message can be initiated. If CRC is not selected, TEOM should not be set.
10	TABORT	BOP	Transmitter Abort = 1 will cause ABORT or FLAG to be sent (IDLE = 0 or 1) after the current character is transmitted (ABORT = 11111111)
11	TGA	BOP	Transmit Go Ahead (GA) instead of FLAG when TEOM = 1. This facilitates repeater termination in loop mode (GA = 01111111)
12-14	Not Defined		
15	TERR	Read only	Transmitter Error = 1 indicates the TxDB has not been loaded in time (one character time $-1/2$ TxC period after TxBE is asserted) to maintain continuous transmission TxU will be asserted to inform the processor of this condition TERR is cleared by setting TSOM. See timing diagram.
		BOP	ABORT's or FLAG's are sent as fill characters (IDLE = 0 or 1)
		BCP	SYNC's or MARK's are sent as fill characters (IDLE = 0 or 1). For IDLE = 1 the last character before underrun is not valid.

Table 7 TRANSMIT DATA/STATUS REGISTER (TDSR) (R/W except TDSR 15) (Cont'd)

BIT	NAME	MODE	FUNCTION
00-07	RxDB	BOP/BCP	Receiver Data Buffer. Contains assembled characters from the RxSR If VRC is specified, the parity bit is stripped.
08	RSOM	BOP	Receiver Start of Message = 1 when a FLAG followed by a non-FLAG has been received and the latter character matches the secondary station address if SAM = 1 RxA will be asserted when RSOM = 1. RSOM resets itself after one character time and has no effect on RxSA.
09	REOM	BOP	Receiver End of Message = 1 when the closing FLAG is detected and the last data character is loaded into RxDB or when an ABORT/GA character is received. REOM is cleared on reading $RDSR_H$, reset operation, or dropping of RxE
10	RAB/GA	BOP	Received ABORT or GA character = 1 when the receiver senses an ABORT character if SS/GA = 0 or a GA character if SS/GA = 1 RAB/GA is cleared on reading $RDSR_H$ reset operation, or dropping of RxE. A received ABORT does not set RxDA
11	ROR	BOP/BCP	Receiver Overrun = 1 indicates the processor has not read last character in the RxDB within one character time $+1/2$ RxC period after RxDA is asserted. Subsequent characters will be lost ROR is cleared on reading $RDSR_H$, reset operation, or dropping of RxE
12-14	ABC	BOP	Assembled Bit Count. Specifies the number of bits in the last received data character of a message and should be examined by the processor when REOM = 1 (RxDA and RxSA asserted). ABC = 0 indicates the message was terminated (by a FLAG or GA) on a character boundary as specified by PCR_{8-10}. Otherwise, ABC = number of bits in the last data character. ABC is cleared when $RDSR_H$ is read, reset operation, or dropping RxE The residual character is right justified in $RDSR_L$.
15	RERR	BOP/BCP	Receiver Error indicator should be examined by the processor when REOM = 1 in BOP, or when the processor determines the last data character of the message in BCP with CRC or when RxSA is set in BCP with VRC. CRC-CCITT preset to 1's/0's as specified by $PCSAR_{8-10}$: RERR = 1 indicates FCS error (CRC ≠ F0B8/ ≠ 0) RERR = 0 indicates FCS received correctly (CRC = F0B / = 0) CRC-16 preset to 0's on 8-bit data characters specified by $PSCAR_{8-10}$ RERR = 1 indicates CRC-16 received correctly (CRC = 0). RERR = 0 indicates CRC-16 error (CRC ≠ 0) VRC specified by $PCSAR_{8-10}$ RERR = 1 indicates VRC error RERR = 0 indicates VRC is correct

Table 8 RECEIVER DATA/STATUS REGISTER (RDSR)-(Read Only)

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER (MPCC) SC2652

ABSOLUTE MAXIMUM RATINGS[1]

	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	0 to +70	°C
T_{STG}	Storage temperature	-65 to +150	°C
	Input or output voltages with respect to GND[3]	-0.3 to +15	V
V_{CC}	With respect to GND	-0.3 to +7	V

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = +5V ±5%[4,5]

	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 1.6mA			0.4	
V_{OH}	High	I_{OH} = -100μA	2.4			
I_{CC}	Power supply current	V_{CC} = 5.25V, T_A = 0°C			150	mA
	Leakage current					μA
I_{IL}	Input	V_{IN} = 0 to 5.25V			10	
I_{OL}	Output	V_{OUT} = 0 to 5.25V			10	
	Capacitance					pF
C_{IN}	Input	V_{IN} = 0V, f = 1MHz			20	
C_{OUT}	Output	V_{OUT} = 0V, f = 1MHz			20	

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 5%[4,5,6]

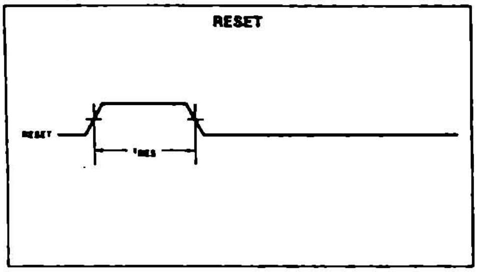
	PARAMETER	2652 Min	2652 Typ	2652 Max	2652-1 Min	2652-1 Typ	2652-1 Max	UNIT
	Setup and hold time							ns
t_{ACS}	Address/control setup	50			50			
t_{ACH}	Address/control hold	0			0			
t_{DS}	Data bus setup (write)	50			50			
t_{DH}	Data bus hold (write)	0			0			
t_{RXS}	Receiver serial data setup	150			150			
t_{RXH}	Receive serial data hold	150			150			
	Pulse width							ns
t_{RES}	RESET	250			250			
t_{DBEN}	DBEN	250		∞[7]	200		∞[7]	
	Delay time							ns
t_{DD}	Data bus (read)			200			170	
t_{TXD}	Transmit serial data			325			250	
t_{DBEND}	DBEN to DBEN delay	200			200			
t_{DF}	Data bus float time (read)			150			150	ns
f	Clock (RxC, TxC) frequency			1.0			2.0	MHz
t_{CLK1}	Clock high (MM = 0)	340			165			ns
t_{CLK1}	Clock high (MM = 1)	490			240			
t_{CLK0}	Clock low	490			240			

NOTES

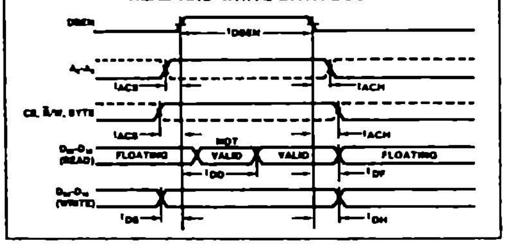
1 Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation sections of this specification is not implied.
2 For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient for ceramic package.
3 This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4 Parameters are valid over operating temperature range unless otherwise specified.
5 All voltage measurements are referenced to ground. All time measurements are at 0.8V or 2.0V. Input voltage levels for testing are 0.4V and 2.4V.
6 Output load C_L = 100pF.
7 m = TxC low and applies to writing to TDSR$_H$ only.

TIMING DIAGRAMS

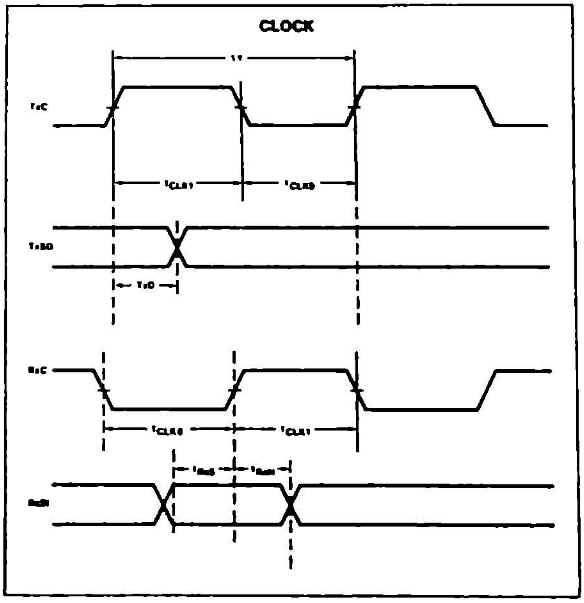
RESET



READ AND WRITE DATA BUS

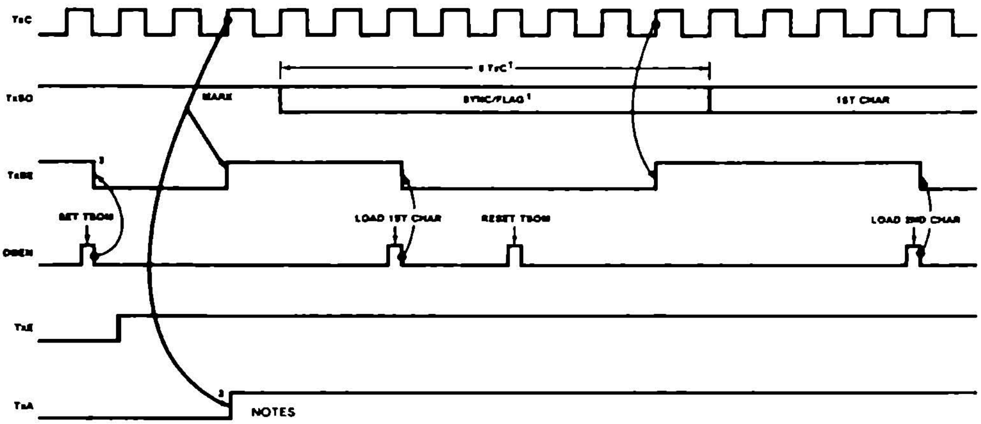

DBEN
tDDEN
A0-A2
tACS
tACH
CE, R/W, BYTE
tACS
tACH
D0-D15 (READ)
FLOATING
NOT VALID
VALID
FLOATING
tDD
tDF
D0-D15 (WRITE)
tDS
tDH

CLOCK


TxC
tT
tCLK1
tCLK0
TxSD
TxD
RxC
tCLK0
tCLK1
tRxS
tRxH
RxSI

TIMING DIAGRAMS (Cont'd)

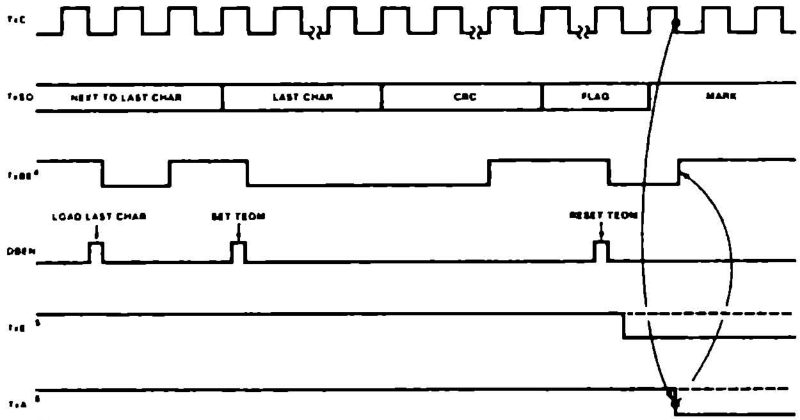
TRANSMIT—START OF MESSAGE


TxC
8 TxC 1
TxSO
MARK
SYNC/FLAG 1
1ST CHAR
TxBE
3
SET TSOM
LOAD 1ST CHAR
RESET TSOM
LOAD 2ND CHAR
DBEN
TxE
TxA
2

NOTES

1 SYNC may be 5 to 8 bits and will contain parity bit as specified
2 TxA goes high relative to TxC rising edge after TSOM has been set and TxE has been raised
3 TxBE goes low relative to DBEN falling edge on the first write transfer into TDSR. It is reasserted 1 TxC time before the first bit of the transmitted SYNC/FLAG. TxBE then goes low relative to DBEN falling edge when writing into $TDSR_H$ and/or $TDSR_L$. It is reasserted on the rising edge of the TxC that corresponds to the transmission of the last bit of each character, except in BOP mode when the CRC is to be sent as the next character (see Transmit Timing—End of Message)

TRANSMIT—END OF BOP MESSAGE

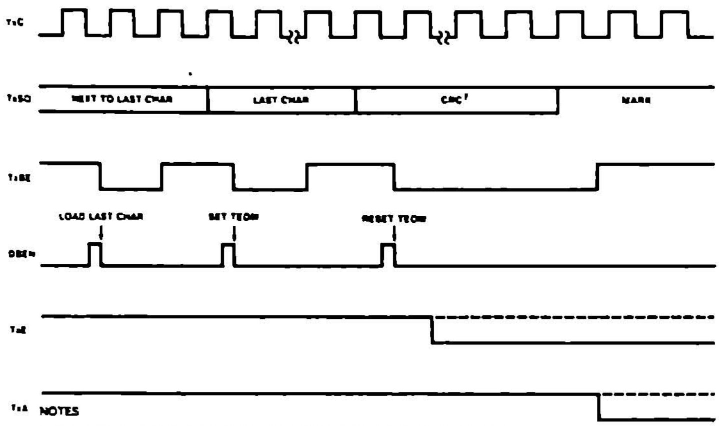

TxC
TxSO
NEXT TO LAST CHAR
LAST CHAR
CRC
FLAG
MARK
TxBE 4
LOAD LAST CHAR
SET TEOM
RESET TEOM
DBEN
TxE 5
TxA 6

NOTES

4 TxBE goes low relative to the falling edge of DBEN corresponding to loading $TDSR_{H/L}$. It goes high one TxC before character transmission begins and also when TxA has been dropped
5 TxE can be dropped before resetting TEOM if TxBE (corresponding to the closing FLAG) is high. Alternatively TxE can remain high and a new message initiated
6 TxA goes low after TxE has been dropped and 1 1/2 TxC's after the last bit of the closing FLAG has been transmitted

TIMING DIAGRAMS (Cont'd)

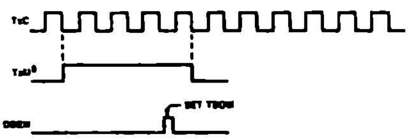
TRANSMIT TIMING—END OF BCP MESSAGE


TxC
TxSO
NEXT TO LAST CHAR
LAST CHAR
CRC 7
MARK
TxBE
LOAD LAST CHAR
SET TEOM
RESET TEOM
DBEN
TxE
TxA

NOTES

7 When 2652 generated CRC is not required, TEOM should only be set if SYNCs are to follow the message block. In that case, TxE should be dropped in response to TxBE (which corresponds to the start of transmission of the last character). When CRC is required, TxE must be dropped before CRC transmission is complete. Otherwise, the contents of TxDB will be shifted out on TxSO. This facilitates transmission of contiguous messages.

TRANSMIT UNDERRUN


TxC
TxU 8
SET TSOM
DBEN

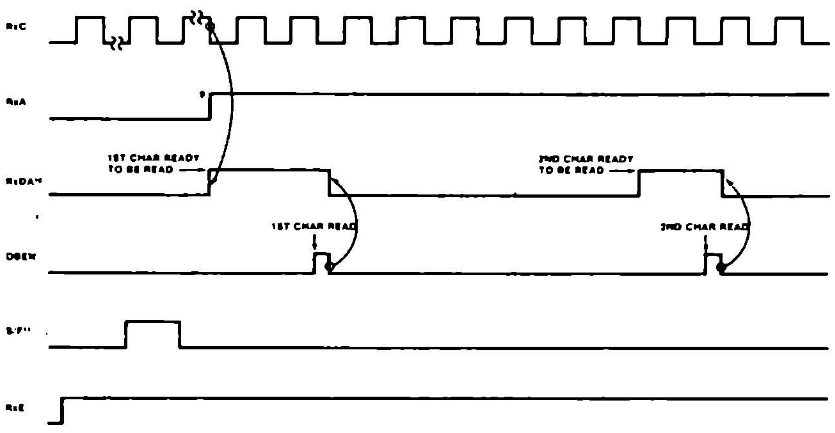
NOTES

8 TxU goes active relative to TxC falling edge if TxBE has not been serviced after n-1/2 TxC times (where n = transmit character length). TxU is reset on the TxC falling edge following assertion of the TSOM command.

9 An underrun will occur at the next character boundary if TEOM is reset and the transmitter remains enabled unless the TSOM command is asserted or a character is loaded into the TxDB.

TIMING DIAGRAMS (Cont'd)

RECEIVE—START OF MESSAGE


RxC
RxA
9
RxDA 10
1ST CHAR READY TO BE READ
2ND CHAR READY TO BE READ
DBEN
1ST CHAR READ
2ND CHAR READ
S/F 11
RxE

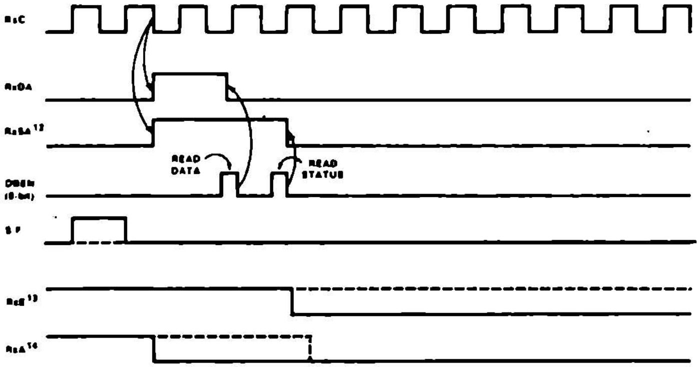
NOTES

9 RxA goes high relative to falling edge of RxC when RxE is high and
  - a A data character following two SYNC's is in RxDB (BCP mode)
  - b Character following FLAG is in RxDB (BOP primary station mode)
  - c Character following FLAG is in RxDB and character matches the secondary station address or All Parties Address (BOP secondary station mode)

10 RxDA goes high on RxC falling edge when a character in RxDB is ready to be read. It comes up before RxSA and goes low on the falling edge of DBEN when RxDB is read

11 S/F goes high relative to rising edge of RxC anytime a SYNC (BCP) or FLAG (BOP) is detected

RECEIVE END OF MESSAGE


RxC
RxDA
RxSA 12
READ DATA
READ STATUS
DBEN (8-bit)
S/F
RxE 13
RxA 14

NOTES

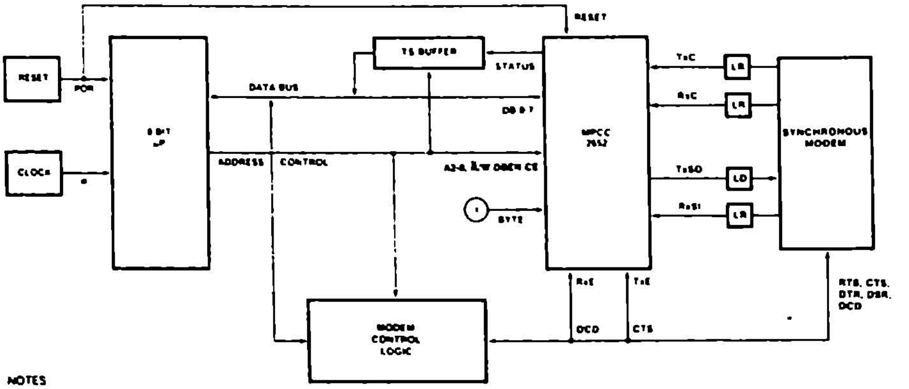
12 At the end of a BOP message, RxSA goes high when FLAG detection (S/F = 1) forces REOM to be set. Processor should read the last data character ($RDSR_L$) and status ($RDSR_H$) which resets RxDA and RxSA respectively. For BCP end of message, RxSA may not be set and S/F = 0. The processor should read the last data character and status

13 RxE must be dropped for BCP with non-contiguous messages. It may be left on at the end of a BOP message (see BOP Receive Operation)

14 RxA is reset relative to the falling edge of RxC after the closing FLAG of a BOP message (REOM = 1 and RxSA active,) or when RxE is dropped

TYPICAL APPLICATIONS

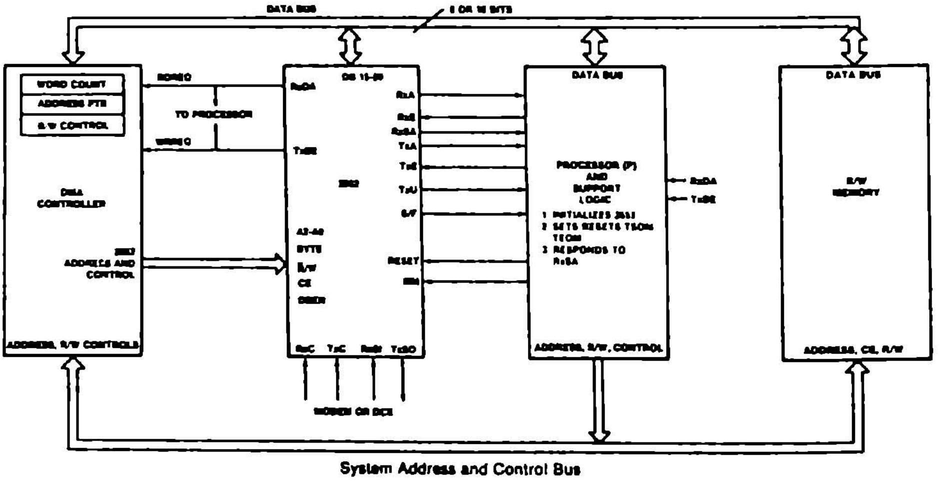
2652 MPCC MICROPROCESSOR INTERFACE


RESET
RESET
POR
CLOCK
Φ
8 BIT
µP
DATA BUS
ADDRESS
CONTROL
TS BUFFER
STATUS
DB 0-7
A2-0, R/W DBEN CE
BYTE
MPCC
2652
TxC
RxC
TxSD
RxSI
LR
LD
SYNCHRONOUS
MODEM
RxE
TxE
DCD
CTS
RTS, CTS,
DTR, DSR,
DCD
MODEM
CONTROL
LOGIC

NOTES

1 Possible µP interrupt requests are
RxDA, RxSA, TxBE, TxU
2 Other 2652 status signals and possible uses are
S/F line, idle indicator, frame delimiter
RxA handshake on RxE, line turn around control
TxA handshake on TxE, line turn around control
3 Line Drivers/Receivers (LD/LR) convert EIA to TTL voltages and vice-versa
4 RTS should be dropped after the CRC (BCP) or FLAG (BOP) has been transmitted
This forces CTS low and TxE low
5 Corresponding high and low order bits of DB must be OR tied

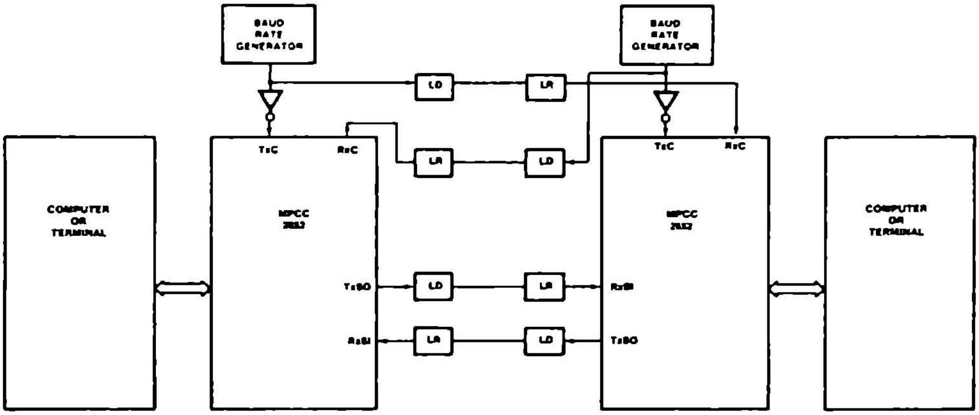
DMA/PROCESSOR INTERFACE


DATA BUS
8 OR 16 BITS
WORD COUNT
ADDRESS PTR
R/W CONTROL
DMA
CONTROLLER
2652
ADDRESS AND
CONTROL
ADDRESS, R/W CONTROLS
RDREQ
WRREQ
TO PROCESSOR
RxDA
TxBE
DB 15-00
2652
RxA
RxE
RxSA
TxA
TxE
TxU
S/F
A2-A0
BYTE
R/W
CE
DBEN
RESET
RxC
TxC
RxSI
TxSD
MODEM OR DCE
DATA BUS
PROCESSOR (P)
AND
SUPPORT
LOGIC
1 INITIALIZES 2652
2 SETS RESETS TSOM
TEOM
3 RESPONDS TO
RxSA
RxDA
TxBE
ADDRESS, R/W, CONTROL
DATA BUS
R/W
MEMORY
ADDRESS, CE, R/W
System Address and Control Bus

For non-DMA operation, TxBE and RxDA are sent to the processor which then loads or reads data characters as required

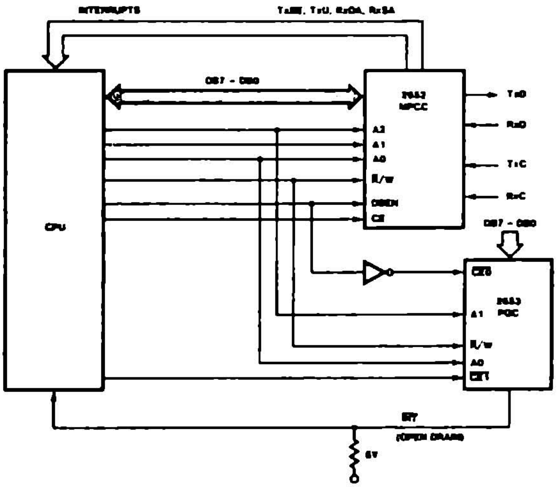
TYPICAL APPLICATIONS (Cont'd)

CHANNEL INTERFACE


BAUD RATE GENERATOR
BAUD RATE GENERATOR
LD
LR
LR
LD
TxC
RxC
TxC
RxC
COMPUTER OR TERMINAL
MPCC 2652
MPCC 2652
COMPUTER OR TERMINAL
TxSO
LD
LR
RxSI
RxSI
LR
LD
TxSO

No Modem—DC Baseband Transmission

2652/2653 INTERFACE
TYPICAL PROTOCOLS: BISYNC, DDCMP, SDLC, HDLC


INTERRUPTS
TxBE, TxU, RxDA, RxSA
DB7 - DB0
CPU
2652 MPCC
A2
A1
A0
R/W
DBEN
CE
TxD
RxD
TxC
RxC
DB7 - DB0
CE0
2653 PGC
A1
R/W
A0
CE1
BY (OPEN DRAIN)
5V

POLYNOMIAL GENERATOR CHECKER (PGC)

SC2653

DESCRIPTION

The Signetics 2653 Polynomial Generator Checker (PGC) is a polynomial generator checker/character comparator circuit that complements a receiver/transmitter (R/T or USART/USRT/UART) in the support of character oriented data link controls. Table 1 defines many of the more commonly used PGC terms and abbreviations.

Parallel data characters transferred between the CPU and R/T are monitored by the PGC which performs block check character (BCC) and parity (VRC) generation/checking, single character detection, and two character sequence detection. Since the PGC operates on parallel characters, the data transmission format may be serial (synchronous or asynchronous) or parallel.

There are four modes of BCC accumulation and each mode can select one of three polynomials to compute the BCC. In the BISYNC normal and transparent modes, the PGC determines which characters are to be accumulated and which characters are to be excluded from the accumulation. The block terminating characters and the initiation and termination of BISYNC transparent text can be detected and an interrupt generated. The single interrupt output represents the inclusive OR of four maskable status conditions.

In the automatic accumulation mode, all characters are accumulated while the single accumulate mode requires a specific accumulation command for each character to be accumulated.

Character accumulation control and character comparisons are facilitated by a character class array which places each of 128 characters into one of four character classes. The four classes are normal, SYN/BISYNC not included, block terminating character (BTC)/search character (SC), and secondary search character (SSC).

Additional PGC applications include off-line R/T operation where the BCC is generated on data not sent to the R/T, BCC multiplexing by sharing the PGC among several R/Ts and reading/writing the partial BCC accumulation on a character by character basis, VRC generation/checking on characters appearing on a bidirectional data bus, and programmable character comparisons or searches.

PGC operation is half duplex (either receive or transmit, one way or two way alternate). Full duplex (two way simultaneous) is achieved by using two PGCs. The device is directly compatible with the Signetics 2651 Programmable Communications Interface (PCI) and 2681 Enhanced Programmable Communications Interface (EPCI). When used in BISYNC modes with the 2681, software requirements are minimized by the 2653-2681 control character comparisons, character sequence comparisons, and automatic DLE insertion/detection.

Other bus oriented R/Ts can be interfaced to the PGC with a minimum of external circuitry. See figure 1 for a typical system configuration.

This NMOS LSI circuit is TTL compatible, operates from a single +5V supply and is contained in a 16 pin dual in line package.

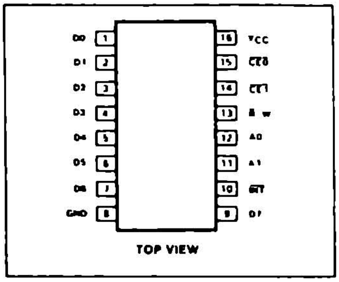
FEATURES

- Parallel Block Check Character accumulation/checking: CRC-16, CRC-12, LRC-8
- BISYNC normal and transparent modes
- Automatic or single character accumulation modes
- Character detection - up to 128 characters
- Two character sequence detection; examples: DLE-STX, ACK 0, ACK 1, WACK, RVI, DISC, WBT
- 6, 7, or 8-bit characters
- VRC generation/checking on data bus
- Four maskable interrupt conditions
- Four classes of characters
- Internal power-on reset
- Maximum character accumulation rate of 500 kHz (4 Mbps)
- Directly compatible with Signetics 2651, 2652 and 2661
- No system clock required
- TTL compatible inputs and outputs
- Single 5V supply
- 16-pin dual in line package

ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ± 5%, T_A = 0°C to 70°C
Ceramic DIP	SC2653CSI16
Plastic DIP	SC2653CSN16

PIN CONFIGURATION



APPLICATIONS

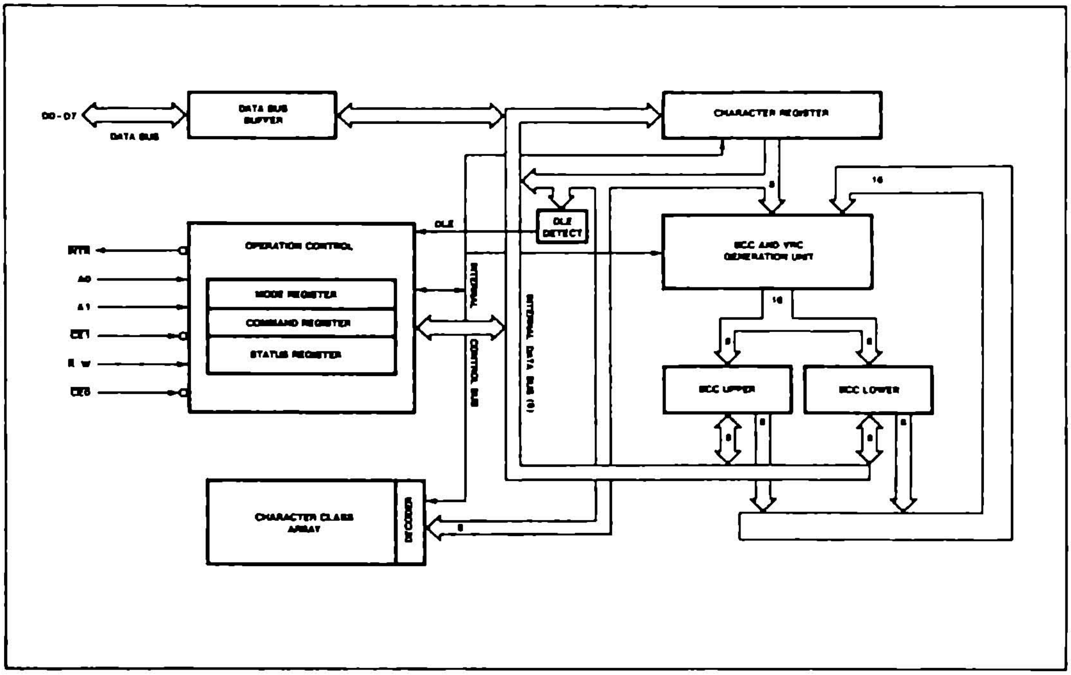
- Character oriented data link control:
  - dedicated to one USART/USRT
  - multiplexed among several USART/USRTs
- Automated BISYNC with 2661 (minimal software intervention)
- BCC and VRC generation/detection on a block of memory or peripheral data
- Programmable character array comparator

BLOCK DIAGRAM

The PGC consists of six major sections. These are the operation control, character class array, DLE ROM, character register, BCC and parity generators, and BCC registers. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the CPU data bus via a data bus buffer.

POLYNOMIAL GENERATOR CHECKER (PGC) SC2653

BLOCK DIAGRAM


D0-D7
DATA BUS
DATA BUS BUFFER
CHARACTER REGISTER
DLE
DLE DETECT
INTR
A0
A1
CE1
R W
CE0
OPERATION CONTROL
MODE REGISTER
COMMAND REGISTER
STATUS REGISTER
INTERNAL CONTROL BUS
INTERNAL DATA BUS (8)
BCC AND VRC GENERATION UNIT
8
16
16
8
8
BCC UPPER
BCC LOWER
8
8
8
8
CHARACTER CLASS ARRAY
DECODER
8

PIN DESIGNATION

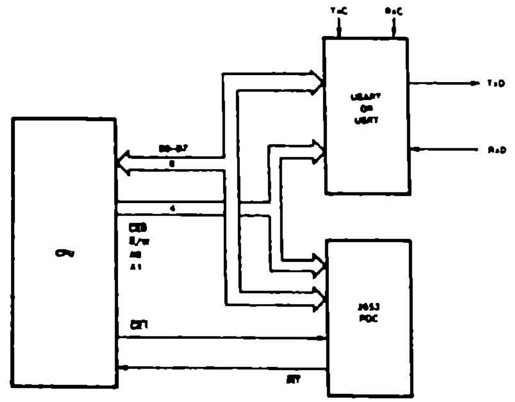
MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
VCC	16	I	+5V: Power supply
GND	8	I	Ground
A1-A0	11,12	I	**Address Lines:** Used to select internal PGC registers or character class array
$\overline{\mathrm{R}}$/W	13	I	**Read/Write:** Read command when low, write command when high
$\overline{\mathrm{CE0}}$	15	I	**Chip Enable:** Connected to chip enable input of a receiver/transmitter (R/T) circuit. It is used to strobe data being transferred between the CPU and the $\overline{\mathrm{R}}$/T into the PGC character register.
$\overline{\mathrm{CE1}}$	14	I	**Chip Enable:** Used in conjunction with the $\overline{\mathrm{R}}$/W signal to enable the transfer of data between the PGC and the CPU or DMA controller and to initialize the PGC registers.
D7-D0	9,7-1	I/O	**Data Bus:** 8-bit three-state bidirectional bus used to transfer data to or from the PGC via $\overline{\mathrm{CE0}}$ or $\overline{\mathrm{CE1}}$. All data, mode words, command words, and status information are transferred on this bus. D0 is the least significant bit; D7 is the most significant bit.
$\overline{\mathrm{INT}}$	10	O	**Interrupt:** Open drain active low interrupt output that signals the CPU that one or more maskable conditions are true: BCC error, VRC error, BTC/SC detect, SSC detect. The true conditions can be determined by reading the status register which in turn deactivates $\overline{\mathrm{INT}}$. A power on, clear BCC, or master reset command causes $\overline{\mathrm{INT}}$ to be inactive (high).

POLYNOMIAL GENERATOR CHECKER (PGC) SC2653

Table 1. GLOSSARY

TERM/ABBREVIATION	DEFINITION
BCC	Block check character
BTC	Block terminating character
SC	Search character
SSC	Second search character (preceded by DLE)
CRC-16	$X^{16} + X^{15} + X^2 + 1$ divisor, dividend pre-cleared
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$ divisor, dividend pre-cleared
LRC-8	Horizontal parity on least significant 7 bits; vertical parity on most significant bit
VRC	Vertical redundancy check (character parity)
R/T	Receiver/transmitter circuit. Also known as USART/USRT/UART/PCI/MPCC
BISYNC	IBM binary synchronous communications (BSC), ANSI X3.28, ISO 1745
MSB	Most significant bit
LSB	Least significant bit
Rx	Receive
Tx	Transmit

TYPICAL SYSTEM CONFIGURATION



NOTES
1. Open drain INT may be OR tied with 2661 PCI, 2661 EPCI, or other open drain interrupt outputs
2. No external circuitry necessary if 2651 or 2661 is the USART

Figure 1

Operation Control Unit

This functional block stores configuration and operation instructions from the CPU and generates appropriate signals to control the device operation. It also contains read and write circuits to permit communications between the CPU and the PGC registers via the data bus. The mode, command, and status registers are in this logic block.

Character Register

Characters to be considered for BCC generation, parity generation and checking, or character comparisons are loaded into this register by either $\overline{CE0}$ or $\overline{CE1}$. This register serves as an input to the BCC and VRC generator, where the accumulation and parity generation takes place. The character register also serves as the input for character class array and DLE comparisons.

Character Class Array

This 128 x 2 array holds the character class associated with each of 128 possible 7-bit characters. The array is zero after a master reset. When the character class array is loaded (see PGC Addressing), the character on the data bus is placed in the class specified by the contents of command register bits CR2 and CR3. The PGC uses these two command bits to represent four different character classes. These are:

1. Normal class (included in the accumulation)
2. SYN character/BISYNC not included class
3. Block terminating character/search class
4. Second search character class (preceded by DLE)

These encoded character classes are used by the PGC:

1. To control the BCC accumulation of associated characters in BISYNC modes only. BCC accumulation in automatic or single accumulation modes is carried out independent of the character classes.
2. To detect characters and two character sequences in all modes of accumulation and to set the control character detect bits in the status register.

It should be noted that any number of characters (up to 128 for CRC-16 or LRC-8; up to 64 for CRC-12) can be put into any one class.

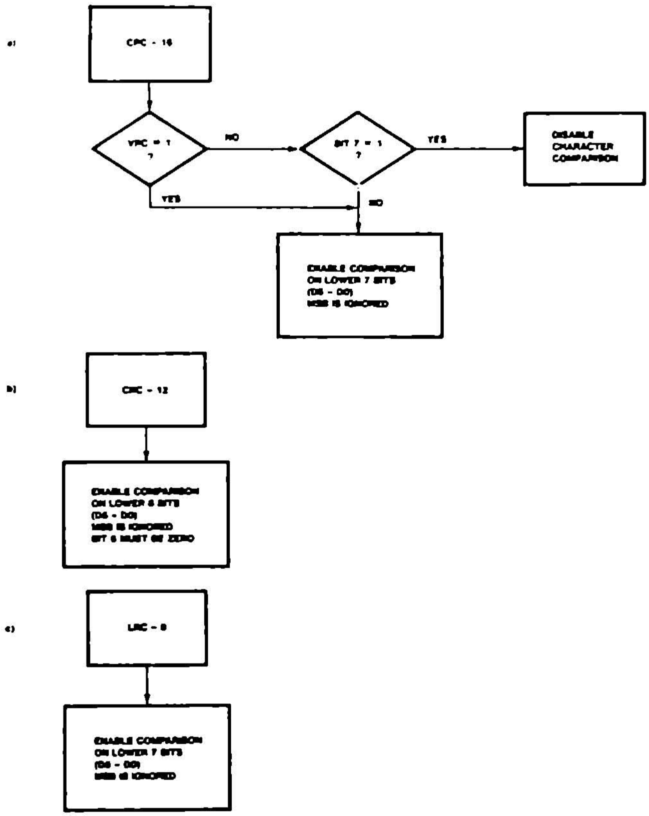
If VRC is specified along with CRC-16 then the least significant 7 bits of the character are used for character array comparison. If VRC is not enabled, but CRC-16 is, the MSB of the character then determines whether a character comparison is to take place. If the MSB is 0, the comparison takes place; if the MSB is 1, the comparison does not take place and the character is processed as though it were in the normal class. This enables the PGC to detect all communication control characters and DLE-SSC sequences.

Only the first 64 locations of the array are accessed if CRC-12 is selected. The user should right justify each six bit character (D0-D5) to be written into the character class array. Bit 6 must be zero.

If VRC is enabled, the generated parity becomes the most significant bit of the character to be compared. VRC is not allowed in BISYNC transparent mode.

The method in which the character register contents is compared against the character class array depends on the BCC polynomial chosen. Figure 2 illustrates the comparison process.

CHARACTER CLASS ARRAY COMPARISON OPERATION



NOTE
Transparent mode always disables the SSC comparison.

Figure 2

DLE Read Only Memory

The DLE characters are stored internally and are selected by the error polynomial as follows:

CRC-12: 01 1111
LCR-8 or CRC-16:
  No VRC or odd VRC. 0001 0000
  Even VRC 1001 0000

BCC and Parity Generator

This functional block performs all the necessary computation to generate and update the BCC accumulation on a character by character basis. It contains the three generator polynomials (CRC-16, CRC-12, and LRC-8) that can be selected to compute the BCC. This block also checks and generates odd or even parity for 7-bit (ASCII) characters.

BCC Registers

This block consists of two 8-bit registers (BCC upper and BCC lower) which contain the high and low order bytes of the BCC accumulation. The result of the accumulation from the BCC and parity generator is stored in these registers. A recirculating register address pointer is initialized by a power on, master reset, or clear BCC command. The pointer alternately selects BCC upper and lower on successive BCC register accesses for CRC-16 or CRC-12. For LRC-8, BCC upper is always selected.

BCC upper and lower are cleared by a clear BCC or master reset command. The highest term of the BCC polynomial is always represented by bit 0 of BCC upper; the lowest term is always represented by bit 7 of BCC lower (see figure 3, Orientation of BCC Polynomials.)

The length of the block check character depends on the error checking polynomial that is selected. If LRC-8 is chosen, the BCC result is stored entirely in BCC upper. The BCC lower remains unchanged from previous setting. Both BCC registers are used when CRC-16 is specified. When CRC-12 is selected, the block check character is 12 bits long. The six least significant bits of the BCC are stored in the least significant bits of the BCC lower. The remaining upper six bits of the BCC are stored in least significant bits of BCC upper. The two most significant bits in each BCC register are filled with zero.

The BCC register(s) are read by the CPU after the last data character is transmitted. They can then be sent to the R/T to complete a transmitted block of data. These registers are read and loaded when one PGC is time-shared by several R/Ts. Refer to Applications Information - Multiplexed PGC.

PGC Addressing

All internal registers and the character class array are selected by the unique address codes shown in table 2.

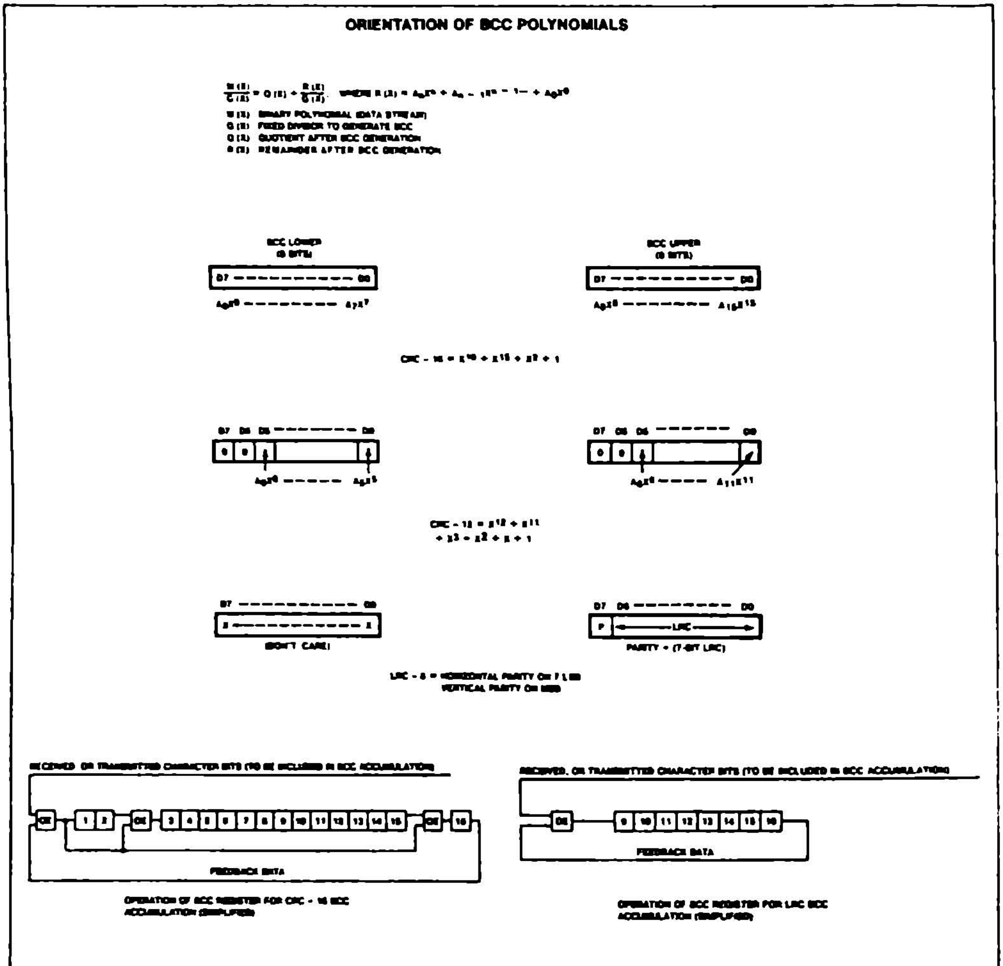

ORIENTATION OF BCC POLYNOMIALS
M(X)/G(X) = Q(X) + R(X)/G(X), WHERE R(X) = AnX^n + An-1X^n-1 — + A0X^0
M(X) BINARY POLYNOMIAL (DATA STREAM)
G(X) FIXED DIVISOR TO GENERATE BCC
Q(X) QUOTIENT AFTER BCC GENERATION
R(X) REMAINDER AFTER BCC GENERATION
BCC LOWER (8 BITS)
D7 D0
A0X0 A7X7
BCC UPPER (8 BITS)
D7 D0
A8X8 A15X15
CRC-16 = X16 + X15 + X2 + 1
D7 D6 D5 D0
0 0
A0X0 A5X5
D7 D6 D5 D0
0 0
A6X6 A11X11
CRC-12 = X12 + X11 + X3 + X2 + X + 1
D7 D0
X X
(DON'T CARE)
D7 D6 D0
P LRC
PARITY + (7-BIT LRC)
LRC-8 = HORIZONTAL PARITY ON 7 LSB VERTICAL PARITY ON MSB
RECEIVED OR TRANSMITTED CHARACTER BITS (TO BE INCLUDED IN BCC ACCUMULATION)
XOR 1 2 XOR 3 4 5 6 7 8 9 10 11 12 13 14 15 XOR 16
FEEDBACK BITS
OPERATION OF BCC REGISTER FOR CRC-16 BCC ACCUMULATION (SIMPLIFIED)
RECEIVED, OR TRANSMITTED CHARACTER BITS (TO BE INCLUDED IN BCC ACCUMULATION)
XOR 9 10 11 12 13 14 15 16
FEEDBACK DATA
OPERATION OF BCC REGISTER FOR LRC BCC ACCUMULATION (SIMPLIFIED)

Figure 3

Table 2. ADDRESS CODES

$\overline{\text{CE0}}$	$\overline{\text{CE1}}$	A1	A0	$\overline{\text{R}}$/W	FUNCTION
0	0	X	X	X	Operation not guaranteed
0	1	0	0	0	If MR2 = 0 load data bus into character register If MR2 = 1 PGC not selected[1]
0	1	0	0	1	If MR2 = 1 load data bus into character register If MR2 = 0 PGC not selected[1]
0	1	0	1	X	PGC not selected[1]
0	1	1	0	X	PGC not selected[1]
0	1	1	1	X	PGC not selected[1]
1	0	0	0	0	Read character register
1	0	0	0	1	Load data bus into character register if MR1,0 ≠ 00[2]; write character class array using CR3, CR2 class code if MR1,0 = 00[3,4]
1	0	0	1	0	Read Status register
1	0	0	1	1	Write command register
1	0	1	0	0	Read mode register
1	0	1	0	1	Write mode register
1	0	1	1	0	Read BCC upper/lower[5]
1	0	1	1	1	Write BCC upper/lower[5]
1	1	X	X	X	PGC not selected[1]

NOTES

1. Data bus in 3-state
2. Character will not be accumulated unless MR3 = 1.
3. Character will not be accumulated even if MR3 = 1.
4. The mode bits MR1 and MR0 are cleared to 00 by power-on-reset, master reset, or by loading the mode register bits MR1 and MR0.
5. Recirculating internal pointer selects BCC Upper on first access, BCC lower on next access for all BCCs except for LRC-8; in case of LRC-8, the pointer only selects BCC upper.

INTERFACE SIGNALS AND TIMING

PGC data transfers are controlled by A1, A0, and $\overline{\text{R}}$/W which must be stable prior to the active low going chip enable pulse. $\overline{\text{CE0}}$ is used for PGC monitoring of data transfers between a CPU/DMA controller and a R/T; $\overline{\text{CE1}}$ is used for direct CPU-to-PGC transfers. MR3 must be set prior to loading the character register in order to accumulate or compare characters via $\overline{\text{CE1}}$. The active low (leading) edge of chip enable initiates a PGC read/write cycle; the rising (trailing) edge ends the cycle and also serves as a write strobe.

When loading the character, mode, or command register, the data bus is strobed into the selected register on the trailing (rising) edge of the appropriate $\overline{\text{CE}}$. When writing into the character class array, the data on the bus (the special character) is placed in the class specified by command register bits CR3 and CR2.

Characters are transferred into the character register when $\overline{\text{CE0}}$ is active (low) depending on the state of MR2 and the $\overline{\text{R}}$/W input. Characters from the R/T are loaded into the character register when in receive mode (MR2 = 0 and $\overline{\text{R}}$/W = 0) while CPU/DMA characters are loaded into the character register when in transmit mode (MR2 = 1 and $\overline{\text{R}}$/W = 1). The time between consecutive chip enables is given by t_{CEC} or t_{CED}.

The open drain active low interrupt signal ($\overline{\text{INT}}$) goes active whenever one or more of four maskable status conditions (SR0-SR3) are true (= 1). A status read deactivates $\overline{\text{INT}}$.

The same techniques used in interfacing the 2651 PCI to 8-bit microprocessors can be used to interface the 2653 PGC (consult Application Note M22). Note that when addressing the R/T's holding registers, the PGC pins must have A1,A0 = 00 and that the address and $\overline{\text{R}}$/W signals must be stable (set up) prior to the active low chip enable. When using the 2651 or 2661 as the R/T, the PGC's A1, A0, $\overline{\text{R}}$/W, and $\overline{\text{CE0}}$ are directly connected to comparable 2651 or 2661 signals. Schematics of a 2653 monitoring data transfers to/from the Signetics 2651/2661 and 2652 are shown in figures 4 and 5.

An alternate interfacing technique is to treat the PGC as an independent peripheral device. This necessitates a write character register instruction after the CPU reads or writes a character to or from the R/T.

2651 OR 2661/2653 INTERFACE

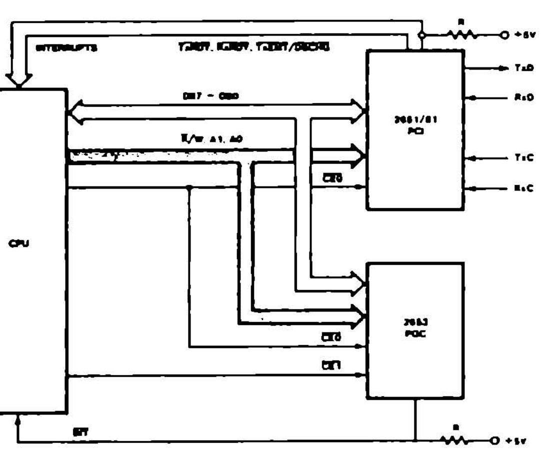


Figure 4

2652/2653 INTERFACE
TYPICAL PROTOCOLS: BISYNC, DDCMP, SDLC, HDLC

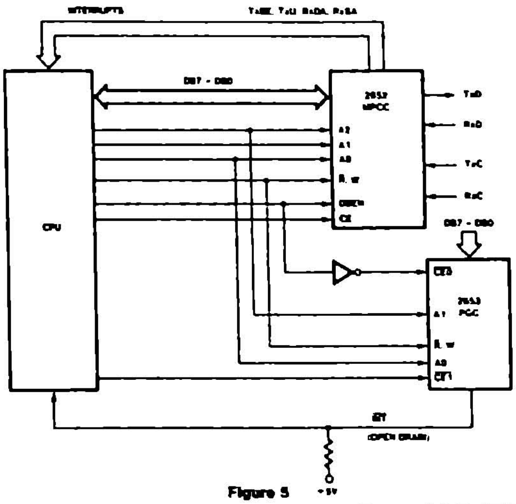


Figure 5

PGC PROGRAMMING

The PGC operational mode must be initially programmed by the CPU (see figure 6). The mode register, command register and character class array should be written into, after a power-on-reset or a master reset command. The character class array should be programmed only for the classes pertinent to the application. After a master reset, the character class array is zero which places all characters in the normal class (included in the BCC accumulation).

OPERATION

The PGC should be initially configured by the CPU (via $\overline{CE1}$) prior to systems operation. This is done by loading the mode register, command register and character class array (see PGC PROGRAMMING). Characters may then be loaded into the character register for BCC accumulation, VRC generation/checking, BTC/SC and DLE-SSC comparisons. See table 3 for a summary of BCC accumulation modes.

BCC accumulation depends on the mode selected.

BISYNC Normal

In BISYNC normal mode, all characters loaded into the character register are accumulated except those in the SYN/BISYNC not included class. During receive (MR2 = 0), a BTC/SC match will cause the BCC accumulation to stop after the next one (LRC-8) or two (CRC-12 or CRC-16) characters have been accumulated. At that time, if the BCC accumulation does not equal zero, the BCC error bit (SR0) will be set and $\overline{INT}$ will go active if the corresponding mask bit (CR4) is enabled (= 1). In transmit (MR2 = 1), the BCC accumulation is automatically stopped once the BTC/SC character has been accumulated. The CPU must read the BCC upper and BCC lower (CRC-12 or CRC-16) register(s) and transmit them to the R/T.

Note that the received BCCs are not subject to VRC if CRC-16 is selected. If LRC-8 is selected, the received BCC is subject to VRC. An incorrect result will set the VRC error bit (SR1). After its accumulation, the least significant 7 bits of BCC upper are checked and a non-zero result will set the BCC error bit (SR0). BCCs are not checked against the character class array nor are they compared to the DLE ROM.

Second search character (SSC) detection is enabled so that a DLE-STX or two character communication control sequence can be detected.

Table 3. SUMMARY OF BCC ACCUMULATION MODES

ACCUMULATION MODES	START ACCUMULATION	STOP ACCUMULATION	CHARACTERS EXCLUDED FROM ACCUMULATION
BISYNC normal and BISYNC transparent	Clear BCC registers command Mode register is loaded with BISYNC or automatic mode Start accumulation command Load BCC registers	After BTC has been detected and received BCC is accumulated After transmitted BTC has been accumulated Single mode is selected	SYN/BISYNC not included class in normal mode DLE-SYN/not included class and first DLE of a DLE non SYN pair in transparent mode. These characters are not excluded if preceded by an odd number of DLEs
Automatic	Same as above	Single mode selected	None
Single	Start accumulation command	After each character has been accumulated	Up to user who must generate start accumulation command for each character to be included

PGC PROGRAMMING

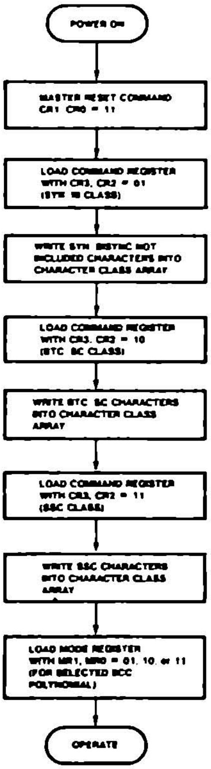
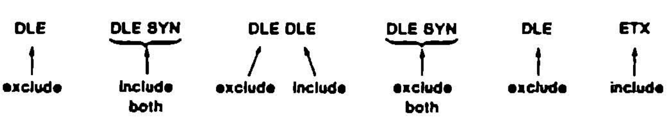

POWER ON
MASTER RESET COMMAND CR1 CR0 = 11
LOAD COMMAND REGISTER WITH CR3, CR2 = 01 (SYN NI CLASS)
WRITE SYN BISYNC NOT INCLUDED CHARACTERS INTO CHARACTER CLASS ARRAY
LOAD COMMAND REGISTER WITH CR3, CR2 = 10 (BTC SC CLASS)
WRITE BTC SC CHARACTERS INTO CHARACTER CLASS ARRAY
LOAD COMMAND REGISTER WITH CR3, CR2 = 11 (SSC CLASS)
WRITE SSC CHARACTERS INTO CHARACTER CLASS ARRAY
LOAD MODE REGISTER WITH MR1, MR0 = 01, 10, or 11 (FOR SELECTED BCC POLYNOMIAL)
OPERATE

Figure 6

BISYNC Transparent

BISYNC transparent mode should be used for data blocks beginning with DLE-STX if the DLEs are transferred between CPU and R/T ($\overline{\text{CE0}}$) or CPU and PGC ($\overline{\text{CE1}}$), i.e., DLEs are *not* stripped. VRC should be disabled in this mode. Characters excluded from the BCC accumulation are the first DLE of a DLE-non SYN sequence pair and the DLE-SYN sequence if not preceded by an odd number of DLEs. For example, consider the following transparent mode character string:


DLE
DLE SYN
DLE DLE
DLE SYN
DLE
ETX
exclude
include both
exclude
include
exclude both
exclude
include

In receive and transmit modes, the termination of BCC accumulation works exactly as in BISYNC normal, except that the BTC/SC must be immediately preceded by an odd number of DLEs to be identified as a BTC/SC.

Second search character detection is not enabled in BISYNC transparent.

After a BTC/SC class character is detected by the PGC when receiving in either BISYNC mode, the following one or two characters are accumulated (depending on LRC-8 or CRC-12/16, respectively) and the PGC will automatically stop further accumulation. However, *the PGC can continue the accumulation* if a start accumulate command is issued or either BISYNC mode is loaded into the mode register. The start accumulate command should be given to the PGC before loading the character that follows the detected BTC/SC. This procedure enables a special search character to be detected (the BTC/SC detect bit (SR2) will be set and an interrupt generated if CR6 = 1) with the BCC accumulation continuing (see figures 7 and 8).

Automatic Accumulate

All characters loaded into the character register are accumulated, BTC/SC and SSC detection is enabled. The BCC accumulation is not automatically terminated. (The CPU must use single accumulate mode to stop the accumulation). When in receive mode, the BCC error bit (SR0) is set/reset after accumulating each character so that the CPU must examine this bit after the last character is accumulated. SR0 = 0 if the accumulated remainder in the BCC register(s) is zero; otherwise SR0 = 1. Examples of use of automatic accumulate mode usage include an R/T (2651/2661) in transparent DLE/SYN strip mode and asynchronous/synchronous/parallel DDCMP.

Single Accumulate

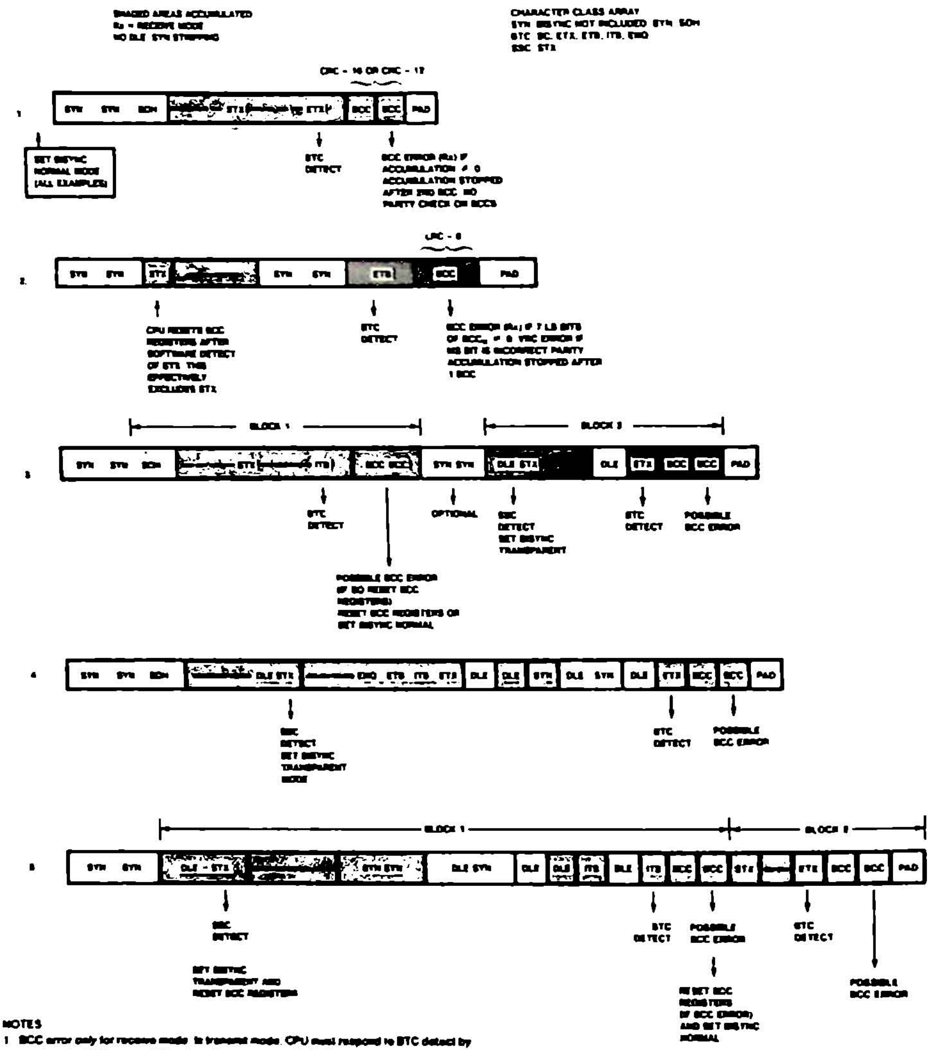
All characters for which a start accumulate command (CR1, CR0 = 01) is given are accumulated and compared against the character class array. If not given, the BCC accumulation is not updated and BTC/SC and SSC detection is disabled. Operation in this mode is otherwise identical to automatic accumulate.

Single accumulate mode can be used to selectively accumulate characters under CPU control or to accumulate characters that were unintentionally excluded in one of the other modes.

Polynomial Selection and DLE Comparison

The BCC polynomial may be CRC-16, CRC-12 or LRC-8. The cyclic redundancy check (CRC) is generated by dividing the binary value of a character in the character register by the selected polynomial. The quotient is discarded and the remainder is used as the BCC (two 8-bit characters for CRC-12, two 8-bit characters for CRC-16). CRC-16 uses all 8 bits of each BCC register. CRC-12 uses the least significant 6 bits of the BCC registers. The two most significant bits of the BCC registers are cleared to zero whenever CRC-12 is selected (see figure 3).

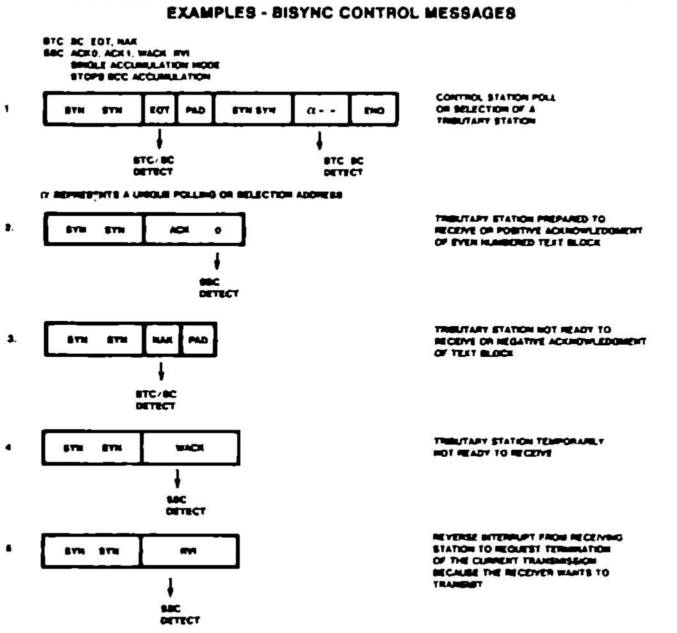
EXAMPLES - BISYNC TEXT MESSAGE BCC ACCUMULATION


SHADED AREAS ACCUMULATED
Rx = RECEIVE MODE
NO DLE SYN STRIPPING
CHARACTER CLASS ARRAY
SYN BISYNC NOT INCLUDED SYN SOH
BTC BC, ETX, ETB, ITB, ENQ
SSC STX
CRC - 16 OR CRC - 12
1
SYN SYN SOH STX ETX BCC BCC PAD
SET BISYNC NORMAL MODE (ALL EXAMPLES)
BTC DETECT
BCC ERROR (Rx) IF ACCUMULATION ≠ 0 ACCUMULATION STOPPED AFTER 2ND BCC. NO PARITY CHECK ON BCCS
LRC - 8
2
SYN SYN STX SYN SYN ETB BCC PAD
CPU RESETS BCC REGISTERS AFTER SOFTWARE DETECT OF STX. THIS EFFECTIVELY EXCLUDES STX
BTC DETECT
BCC ERROR (Rx) IF 7 LS BITS OF BCC ≠ 0. VRC ERROR IF MS BIT IS INCORRECT PARITY. ACCUMULATION STOPPED AFTER 1 BCC
BLOCK 1
BLOCK 2
3
SYN SYN SOH STX ITB BCC BCC SYN SYN DLE STX DLE ETX BCC BCC PAD
BTC DETECT
OPTIONAL
SSC DETECT SET BISYNC TRANSPARENT
BTC DETECT
POSSIBLE BCC ERROR
POSSIBLE BCC ERROR IF NO RESET BCC REGISTERS. RESET BCC REGISTERS OR SET BISYNC NORMAL
4
SYN SYN SOH DLE STX ENQ ETB ITB ETX DLE DLE SYN DLE SYN DLE ETX BCC BCC PAD
SSC DETECT SET BISYNC TRANSPARENT MODE
BTC DETECT
POSSIBLE BCC ERROR
BLOCK 1
BLOCK 2
5
SYN SYN DLE STX SYN SYN DLE SYN DLE DLE ITB DLE ITB BCC BCC STX ETX BCC BCC PAD
SSC DETECT
SET BISYNC TRANSPARENT AND RESET BCC REGISTERS
BTC DETECT
POSSIBLE BCC ERROR
RESET BCC REGISTERS (IF BCC ERROR) AND SET BISYNC NORMAL
BTC DETECT
POSSIBLE BCC ERROR

NOTES

1. BCC error only for receive mode. In transmit mode, CPU must respond to BTC detect by reading the BCC register(s) and sending them to the R/T. The accumulation is stopped after the BTC is accumulated.
2. ENQ (DLE ENQ) in a text message should be treated as an abort.
3. Opening SYNs may be stripped by the R/T.
4. The single accumulate mode and command can be used to accumulate a character that inadvertently was excluded. (For example, the DLE of a DLE STX if the PGC was in transparent mode and there was not a line turnaround prior to the DLE.) The single accumulation should be done using $\overline{CE}$1 after the BCC(s) have been accumulated.

Figure 7


EXAMPLES - BISYNC CONTROL MESSAGES
BTC SC EOT, NAK
SSC ACK0, ACK1, WACK RVI
SINGLE ACCUMULATION MODE
STOPS BCC ACCUMULATION
1
SYN SYN
EOT
PAD
SYN SYN
α - -
ENQ
CONTROL STATION POLL OR SELECTION OF A TRIBUTARY STATION
BTC/SC DETECT
BTC SC DETECT
α REPRESENTS A UNIQUE POLLING OR SELECTION ADDRESS
2.
SYN SYN
ACK 0
TRIBUTARY STATION PREPARED TO RECEIVE OR POSITIVE ACKNOWLEDGMENT OF EVEN NUMBERED TEXT BLOCK
SSC DETECT
3.
SYN SYN
NAK
PAD
TRIBUTARY STATION NOT READY TO RECEIVE OR NEGATIVE ACKNOWLEDGMENT OF TEXT BLOCK
BTC/SC DETECT
4
SYN SYN
WACK
TRIBUTARY STATION TEMPORARILY NOT READY TO RECEIVE
SSC DETECT
5
SYN SYN
RVI
REVERSE INTERRUPT FROM RECEIVING STATION TO REQUEST TERMINATION OF THE CURRENT TRANSMISSION BECAUSE THE RECEIVER WANTS TO TRANSMIT
SSC DETECT

NOTES

1 BCC accumulation should be ignored for control messages. This can be effected by single accumulate mode without single accumulate commands.

2 Characters programmed as SSCs should be the binary equivalent of the second character of the DLE-SSC sequence.

Figure 8

When the PGC is in receive mode (MR2 = 0), the received BCC will be accumulated. The result will be zero for an error free message.

CRC-12 is used with 6-bit codes. The internal 6-bit transcode DLE character hex 1F is selected by CRC-12. VRC should be disabled (MR4 = 0) for CRC-12 operation. The two most significant bits of the character register are ignored when compared to the internal 6-bit DLE. When the character is checked against the character class array, the MSB is ignored and the next MSB (bit 6) is assumed to be zero. If CRC-12 is specified, the user must write to the character class array with bit 6 cleared.

CRC-16 or LRC-8 implies the use of ASCII or EBCDIC although any 7-bit plus parity or 8-bit no parity code may be used (with DLE = hex 10 or hex 90). The DLE character compare is on an 8-bit basis with the generated parity (if VRC is enabled) as the MSB. When the character is compared against the character class array, the MSB is not used. This may result in a false BTC or SSC detection if there is a VRC error. However, the VRC error bit (SR1) will be set under that condition.

The LRC-8 is generated by the exclusive OR of the 7 least significant bits of the character register and the BCC upper. The most significant bit of the LRC-8 check character is a vertical odd/even parity bit (MR5 = 0/1), which is generated on the least significant bits of that character. The selection of LRC-8 implies VRC is enabled and that only the BCC upper is used for the BCC accumulation. The BCC lower remains unchanged from previous setting.

VRC Generation and Detection

Parity (VRC) is enabled by MR4 and specified as odd or even by MR5. VRC should be disabled when in BISYNC transparent mode and whenever CRC-12 or CRC-16 (EBCDIC) is selected as the BCC polynomial. MR4 = 1 enables VRC generation and detection for *both* receive and transmit operations. Characters loaded into the character register will have VRC generated on the least significant 7 bits with the generated parity bit written into the character register MSB. If the generated parity does not match the MSB of the loaded character, the VRC error bit (SR1) is set and $\overline{\text{INT}}$ asserted if the corresponding mask bit was enabled (CR5 = 1). Thus, if 7-bit characters are to be transmitted with VRC, CR5 should be zero and SR1 ignored. 8-bit characters with a VRC bit in the MSB position are parity checked by the PGC in both transmit (to R/T) and receive (from R/T) modes, i.e., the PGC operates as a data bus parity checker.

CHARACTER CLASSES

Normal (Included in the Accumulation)

Any character that belongs to this class is normal data, i.e., the character is not a communication control or other special character. Characters in this class are always accumulated in BISYNC, automatic and single accumulation modes.

SYN Character/BISYNC Not Included

SYN characters are never accumulated in BISYNC normal accumulation mode. In BISYNC transparent accumulation mode, the DLE-SYN character pair is not accumulated, but a SYN not preceded by a DLE is accumulated. (DLE is implied as an odd number of DLEs).

Block Terminating Character (BTC)/Search Character (SC)

BTC/SC characters have two functions in the PGC: termination of BCC accumulation and character detection. In BISYNC transparent mode, a BTC/SC must be preceded by an odd number of DLEs to be recognized.

- **Termination of BCC Accumulation**

  In BISYNC normal and transparent accumulation modes, the PGC will stop the accumulation upon the detection of the BTC/SC character. Examples of BTCs are ETX, ETB, ITB, ENQ.

  In receive mode, the accumulation is stopped after the following one (LRC-8) or two (CRC-12, CRC-16) character(s) have been accumulated. In transmit mode, the accumulation is stopped after the BTC/SC character has been accumulated. The BTC/SC character is always accumulated in all of the accumulation modes.

- **Character Detection**
  BTC/SC characters will be detected in any of the four accumulation modes when that character is being accumulated. The BTC/SC status bit (SR2) is set on detection. Since detection also stops BISYNC BCC accumulation, the BISYNC accumulation must be restarted if the character is not a BTC. This can be effected by loading BISYNC mode into the mode register or generating a start accumulation command.

Second Search Character Class (SSC)

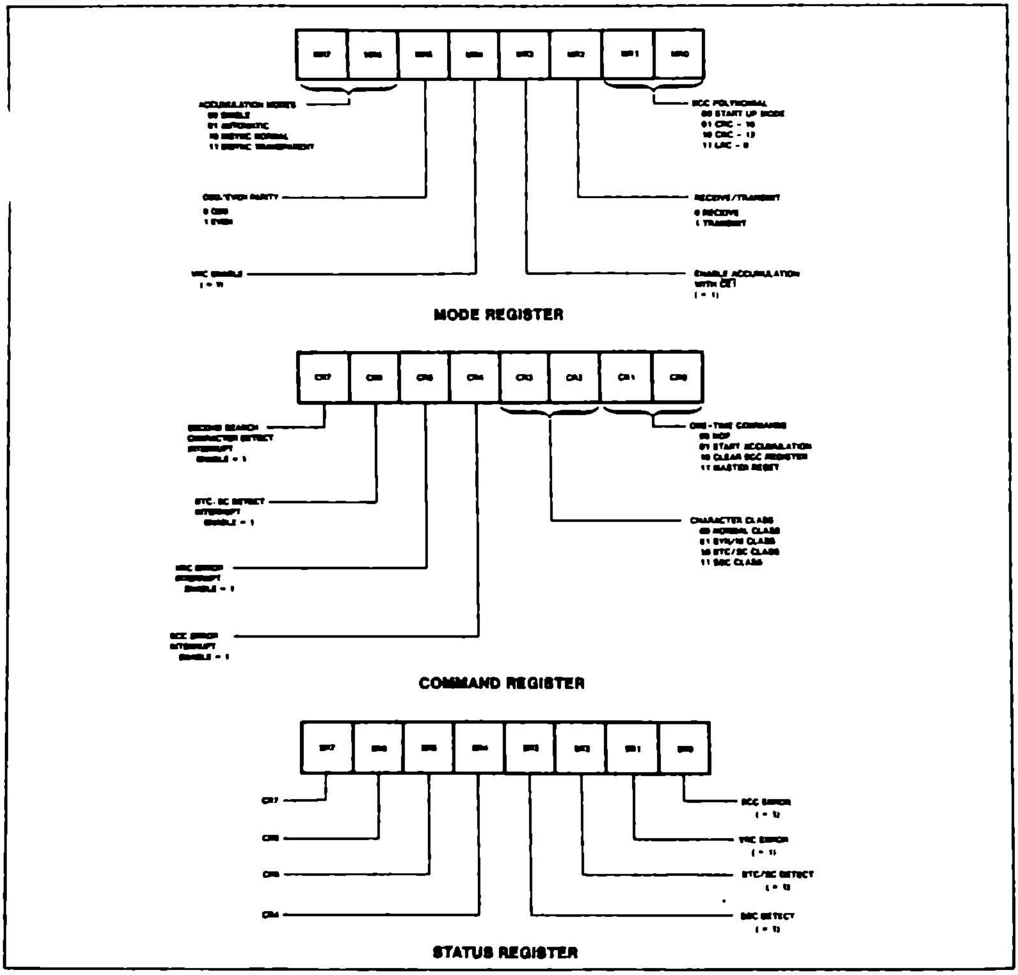
Control functions in character oriented data link control procedures can be represented by a sequence of two characters, the first character being a DLE. Examples include ACK0, ACK1, WACK, RVI, DISC, WBT and the initiation of transparent text (DLE-STX). The PGC will detect such sequences, except in BISYNC transparent mode, when an SSC class character is being accumulated after being immediately preceded by an odd number of DLEs. Under those conditions, the SSC status bit (SR3) will be set.

The SSC character is always accumulated in all of the accumulation modes.

REGISTER BIT DESCRIPTION

The operation of the PGC is determined by programming the mode register and the command register. The status register provides feedback on potential interrupt conditions. Formats of these registers are shown in table 4.

Table 4. PGC REGISTER BIT FORMATS


MR7
MR6
MR5
MR4
MR3
MR2
MR1
MR0
ACCUMULATION MODES
00 SINGLE
01 AUTOMATIC
10 BISYNC NORMAL
11 BISYNC TRANSPARENT
ODD/EVEN PARITY
0 ODD
1 EVEN
VRC ENABLE
(= 1)
BCC POLYNOMIAL
00 START UP MODE
01 CRC - 16
10 CRC - 12
11 LRC - 8
RECEIVE/TRANSMIT
0 RECEIVE
1 TRANSMIT
ENABLE ACCUMULATION
WITH CE1
(= 1)
CR7
CR6
CR5
CR4
CR3
CR2
CR1
CR0
SECOND SEARCH
CHARACTER DETECT
INTERRUPT
ENABLE = 1
BTC, SC DETECT
INTERRUPT
ENABLE = 1
VRC ERROR
INTERRUPT
ENABLE = 1
BCC ERROR
INTERRUPT
ENABLE = 1
ONE-TIME COMMANDS
00 NOP
01 START ACCUMULATION
10 CLEAR BCC REGISTER
11 MASTER RESET
CHARACTER CLASS
00 NORMAL CLASS
01 SYN/ID CLASS
10 BTC/SC CLASS
11 SSC CLASS
SR7
SR6
SR5
SR4
SR3
SR2
SR1
SR0
CR7
CR6
CR5
CR4
BCC ERROR
(= 1)
VRC ERROR
(= 1)
BTC/SC DETECT
(= 1)
SSC DETECT
(= 1)

MODE REGISTER

COMMAND REGISTER

STATUS REGISTER

Table 5. BCC ACCUMULATION BY CHARACTER CLASS

CR3	CR2	CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTOMATIC ACCUM	SINGLE ACCUM
0	0	Normal	Yes	Yes	Yes	Yes
0	1	SYN/BISYNC not included	No	Yes, unless preceded by an odd number of DLEs	Yes	Yes
1	0	BTC/SC	Yes	Yes	Yes	Yes
1	1	SSC*	Yes	Yes	Yes	Yes

NOTE

*Preceded by DLE

Mode Register

The mode register defines general PGC operation characteristics. MR1 and MR0 = 00 permit the character class array to be programmed. These bits will be zero after a power on or master reset command. After the character class array is programmed, these bits should be set to 01, 10, or 11 to select the CRC-16, CRC-12 or LRC-8 polynomials.

MR2 ($T_x/\overline{R}_x$) determines whether or not the PGC is to generate (Tx) or generate and check (Rx) the BCC. It is used with $\overline{R}/W$ to determine if the data bus is to be loaded into the character register when $\overline{CE0}$, $\overline{CE1}$, A1, A0 = 0100.

If MR2 = 1: 1) the PGC will generate the BCC, but will never set the BCC error bit (SR0). 2) If the $\overline{R}/W$ pin is high when $\overline{CE0}$, $\overline{CE1}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{R}/W$ is low under these conditions, the PGC is not selected.

If MR2 = 0: 1) the PGC will accumulate the BCC and set the BCC error bit (SR0) when appropriate. 2) If the $\overline{R}/W$ pin is low when $\overline{CE0}$, $\overline{CE1}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{R}/W$ is high under these conditions, the PGC is not selected.

MR3 is a $\overline{CE1}$ accumulate/compare enable bit. If MR3 = 0, characters loaded into the character register by $\overline{CE1}$ are *not* accumulated, checked against the character class array, or compared to the DLE ROM. Parity will be generated and checked if VRC is enabled (MR4 = 1). The primary use of MR3 = 0 is to generate parity on a 7-bit character which is to be transmitted to an R/T. The CPU loads the character register with the 7-bit character and reads the 8-bit VRC generated character via $\overline{CE1}$. This 8-bit character is then transferred to the R/T via $\overline{CE0}$. Another application of MR3 = 0 is for a CPU to interleave parity checking on memory data ($\overline{CE1}$) with on line R/T data transfers ($\overline{CE0}$).

If MR3 = 1, characters loaded into the character register by $\overline{CE1}$ will be accumulated (according to the BCC accumulation mode selected) and compared against the character class array and DLE ROM. This bit setting should be used when the CPU/DMA controller sends data characters to be accumulated or compared to the PGC and the R/T is inactive (off line). If the R/T were active, then a DLE or BTC loaded into the character register via $\overline{CE0}$ would cause incorrect accumulation and character comparisons if the next character was loaded via $\overline{CE1}$.

MR4 is a VRC enable bit. If MR4 = 1, VRC is enabled as odd/even by MR5. VRC is generated on the 7 LS bits of the character and the MS bit is checked against the generated parity. If not equal, SR1 is set. If MR4 = 0, VRC is not enabled. MR4 = 0 is used for BISYNC transparent mode with ASCII code, and for both BISYNC modes for EBCDIC and SBT.

MR5 is an odd/even VRC bit. If MR5 = 1, the total number of 1 bits in the character including the parity bit is even. If MR5 = 0, the total number of bits is odd. This bit is ignored if MR4 = 0.

MR7, MR6 select the BCC accumulation mode. These modes have been previously discussed in the operation section.

Command Register

The command register contains four interrupt enables, a 2-bit character class code used when programming the character class array, and 2 bits that specify three one time commands and a NOP.

CR1, CR0 = 00 is a NOP. This bit setting is used when changing CR7-CR2 without affecting any of the 3 one time commands.

CR1, CR0 = 01 is a start BCC accumulation command. In single accumulation mode, the character accumulated is the character that is in the character register at the time the command is given. The accumulation stops immediately after the character has been accumulated. If the command is given in either of the BISYNC or automatic accumulation modes, it enables the PGC to accumulate the BCC starting with the *next* character loaded into the character register. This is a means of restarting a BISYNC normal accumulation after detection of a BTC/SC that is not a valid BTC (example; CR, LF, TAB). In all accumulation modes, a previously detected DLE will not be cancelled by this command.

CR1, CR0 = 10 is a clear BCC registers command. Both BCC registers are cleared along with the associated internal pointer and SR0-SR3. The pointer points to BCC upper. $\overline{INT}$ is forced high. This command permits BCC accumulation, starting with the next character loaded into the character register in BISYNC or auto modes. Single accumulate mode requires a start BCC accumulation command.

CR1, CR0 = 11 is a master reset command. All internal registers (except the character register), the internal pointer, and the entire character class array are cleared. $\overline{INT}$ is forced high.

CR3 and CR2 are used for programming the character class array. During a write character class array instruction, the character corresponding to the 7 LS bits of the data bus is placed in the class contained in CR3 and CR2. The encoded character classes control the accumulation of the associated character as shown in table 5.

Detection operates under the conditions shown in table 6.

Table 6. BTC/SC AND SSC DETECTION CONDITIONS

CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTO ACCUM	SINGLE ACCUM
BTC/SC	Yes	Yes*	Yes	Yes †
SSC	Yes*	No	Yes*	Yes*†

NOTES

* Only if immediately preceded by an odd number of DLEs

† Start accumulate command necessary for detection

CR7, CR6, CR5, CR4 are interrupt enables that individually enable/disable $\overline{INT}$ when the corresponding status register condition is true (set). Each bit is set in order to enable $\overline{INT}$ upon the condition. Each bit is reset to disable $\overline{INT}$ upon the condition. The state of these bits may be read via the status register (SR7, SR6, SR5, SR4).

The corresponding status bits (SR3, SR2, SR1, SR0) are set independent of the interrupt enables. The bit assignments are:

CR4 - BCC error interrupt enable

CR5 - VRC error interrupt enable

CR6 - BTC/SC detect interrupt enable

CR7 - DLE-SSC detect interrupt enable

Status Register

This register reflects the status of the 4 conditions that are potential interrupt ($\overline{INT}$) sources and the 4 interrupt enables in the command register. A status register read clears SR0, SR1, SR2, SR3 and deactivates $\overline{INT}$. These bits are also cleared by a master reset or clear BCC command.

SR0 is a BCC error bit. This bit can only be set in receive mode (MR2 = 0). In BISYNC normal and BISYNC transparent modes, SR0 will be set/reset once the accumulation has been stopped by the detection of the BTC/SC character and accumulation of the BCC(s).

In automatic and single accumulate modes, SR0 is set/reset after each character in the character register has been accumulated.

The rules for the detection of a BCC error are:

SR0 = 1 LRC-8: 7 least significant bits of BCC upper ≠ 0
CRC-12, CRC-16: BCC upper *or* BCC lower ≠ 0

SR0 = 0 LRC-8: 7 least significant bits of BCC upper = 0
CRC-12, CRC-16: BCC upper *and* BCC lower = 0

SR1 is a VRC error bit. When set, this bit reports a character parity error (on receive or transmit) when parity is enabled (MR4 = 1). Parity is odd/even as specified by MR5. The parity bit will be regenerated in the character register.

SR2 is a BTC/SC detect bit. When set, this bit indicates the character being accumulated is of the BTC/SC class for BISYNC normal, automatic and single accumulate modes. In BISYNC transparent mode, the BTC/SC character being accumulated must be *immediately* preceded by an odd number of DLEs for this bit to be set.

SR3 is a DLE SSC detect bit. This bit can only be set when in BISYNC normal, auto, or single accumulate modes. When set, it indicates that the character being accumulated is of the SSC class when that character was *immediately* preceded by an odd number of DLEs.

SR7, SR6, SR5, SR4 are interrupt enables. These 4 bits reflect the state of the interrupt enable command bits CR7, CR6, CR5, and CR4, as follows:

SR4 - BCC error

SR5 - VRC error

SR6 - BTC/SC detect

SR7 - SSC detect

APPLICATIONS INFORMATION

Dedicated PGC

The most efficient use of the 2653 is to dedicate one to each R/T for two way alternate (half duplex) operation or two to each R/T for two way simultaneous (full duplex) operation (see figure 9). The CPU configures each PGC (using $\overline{CE1}$) by initializing the mode register, command register, and character class array. Data transfers to or from the R/T can then be on a DMA basis with each receiver holding register ready signal used as a read request (RREQ) and each transmit holding register available signal used as a write request (WREQ) to the DMA controller. The CPU needs only to respond to enabled interrupts from each of the PGCs. The individual $\overline{INT}$ outputs can be wire-OR'd into a single CPU interrupt ($\overline{INTRPT}$) with one pull up resistor. Each PGC in this system has a unique address that is decoded into the respective chip enables.

The CPU or DMA controller could send a block of memory data to the PGC to be error checked without sending that data to the R/T. In that case, $\overline{CE1}$ is used.

Multiplexed PGC

One PGC may be time-shared among a few R/Ts if the CPU saves and restores the mode register and partial BCC result in the BCC registers. These registers are accessed via $\overline{CE1}$. There must be a separate save area for each R/T (serial channel) and a channel pointer indicating the last R/T that transferred or received a data character (see figure 10).

The loading of the BCC registers will clear SR0-SR3 and all previously detected special characters, i.e., DLE, BTC/SC, BCC (BISYNC modes). The BCC accumulation will start again when the next character is loaded into the character register in all accumulation modes except single. That mode requires a start accumulation command.

Figures 11 and 12 represent software flow diagrams for transmit and receive service requests. Note that interrupts from all other R/Ts must be masked during a read or write to the BCC registers so as not to affect the internal BCC address pointer. It is recommended that all R/T interrupts be masked while servicing an interrupt that accesses any PGC register.

BISYNC Operation

Table 7 is a concise listing of 2651/2661 operating modes with recommended corresponding 2653 BCC accumulation modes.

Character Comparator

The PGC can be used as a programmable data bus character comparator which monitors data bus transfers (CPU→peripheral, CPU→CPU, CPU→memory, memory→peripheral (via DMA)). The user selectively loads the character class array with BTC/SC and SSC characters to be compared. Status bits will be set and an interrupt can be generated upon SC and DLE - SSC detection. A match on one to 128 different characters or DLE - SSC sequences can be programmed.

Figure 13 depicts an arrangement where the DMA controller or slave CPU handles data bus transfers, the PGC interrogates the data bus, and the host CPU responds to PGC interrupts.

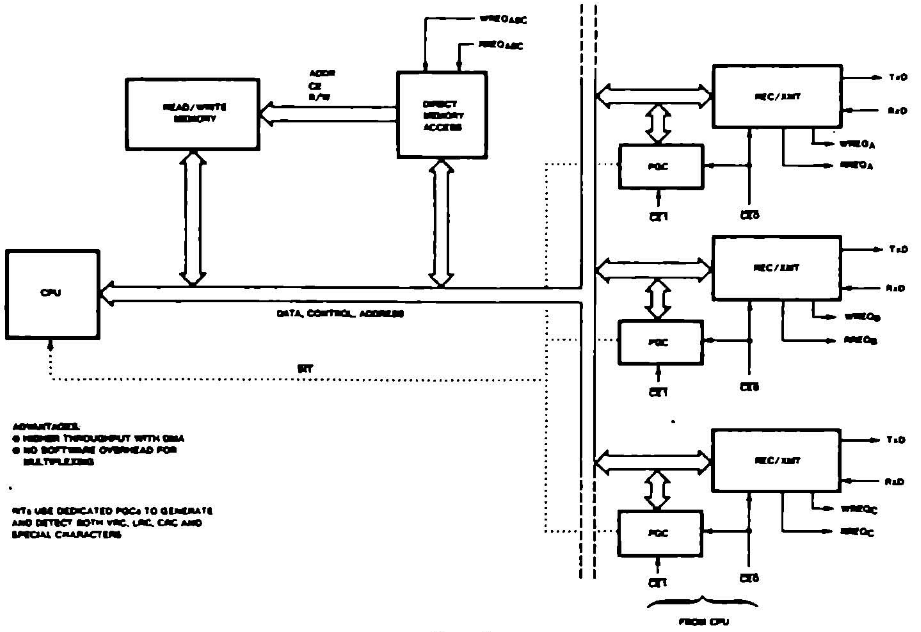
Table 7. BISYNC (ANSI 3.28, ISO 1745) Modes for 2651/2661 and 2653

2651/2661 OPERATING MODES	2653 BCC ACCUMULATION MODE
Sync normal non-strip	BISYNC normal
Sync transparent non-strip	BISYNC transparent
Normal SYN/DLE strip[1]	BISYNC normal
Transparent SYN/DLE strip[1]	Automatic accumulate[2]
Async (with SYN/DLE characters)	BISYNC normal

NOTES

[1] CPU should switch to non-strip mode after BTC detect. Otherwise a received BCC could be inadvertently stripped

[2] SSC detect should be ignored



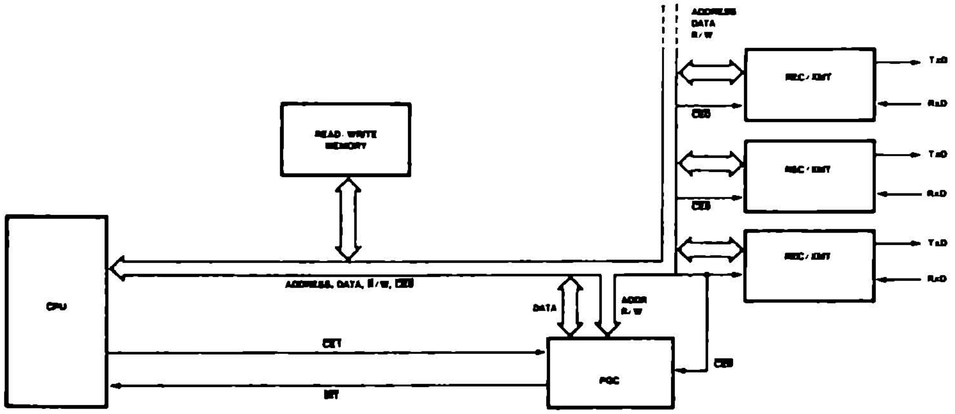
ONE PGC PER RECEIVER/TRANSMITTER

WREQABC
RREQABC
ADDR
CE
R/W
READ/WRITE MEMORY
DIRECT MEMORY ACCESS
CPU
DATA, CONTROL, ADDRESS
INT
REC/XMIT
TxD
RxD
WREQA
RREQA
WREQB
RREQB
WREQC
RREQC
PGC
CE1
CE0
FROM CPU

ADVANTAGES:
- HIGHER THROUGHPUT WITH DMA
- NO SOFTWARE OVERHEAD FOR MULTIPLEXING

R/Ts USE DEDICATED PGCs TO GENERATE AND DETECT BOTH VRC, LRC, CRC AND SPECIAL CHARACTERS

Figure 9



PGC SERVICES MULTIPLE RECEIVERS/TRANSMITTERS

ADDRESS
DATA
R/W
REC/XMIT
TxD
RxD
CE0
READ/WRITE MEMORY
CPU
ADDRESS, DATA, R/W, CE0
DATA
ADDR
R/W
CE1
INT
PGC
CE0

ONE PGC IS TIME-SHARED BY THREE R/Ts.
THE CPU READS AND RESTORES THE PARTIAL BCC REMAINDER FOR EACH SERIAL CHANNEL.

Figure 10

MULTIPLEXED PGC - TRANSMIT TO R/T

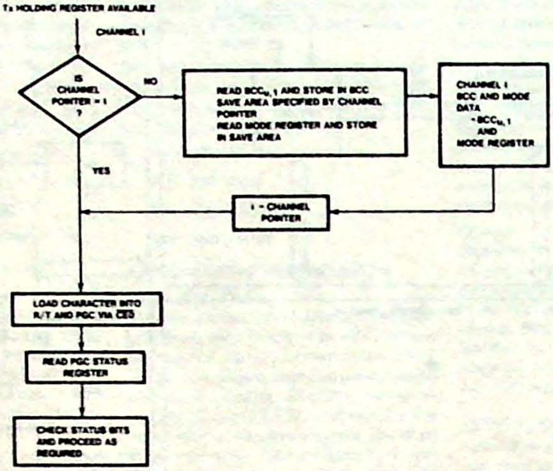

Tx HOLDING REGISTER AVAILABLE
CHANNEL i
IS CHANNEL POINTER = i ?
NO
READ $BCC_{u,i}$ AND STORE IN BCC SAVE AREA SPECIFIED BY CHANNEL POINTER
READ MODE REGISTER AND STORE IN SAVE AREA
CHANNEL i BCC AND MODE DATA - $BCC_{u,i}$ AND MODE REGISTER
YES
i - CHANNEL POINTER
LOAD CHARACTER INTO R/T AND PGC VIA $\overline{CEB}$
READ PGC STATUS REGISTER
CHECK STATUS BITS AND PROCEED AS REQUIRED

Figure 11

MULTIPLEXED PGC - RECEIVE FROM R/T

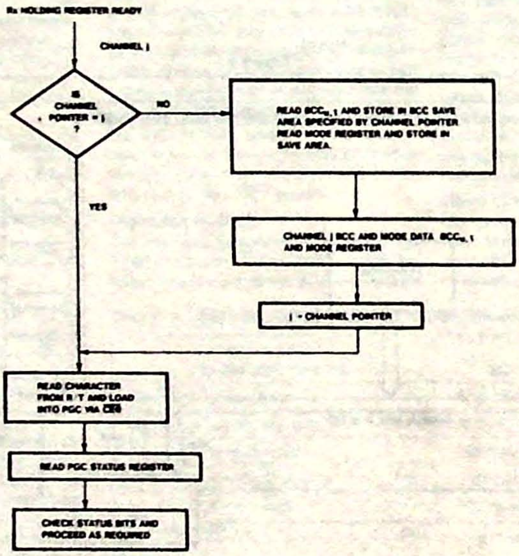
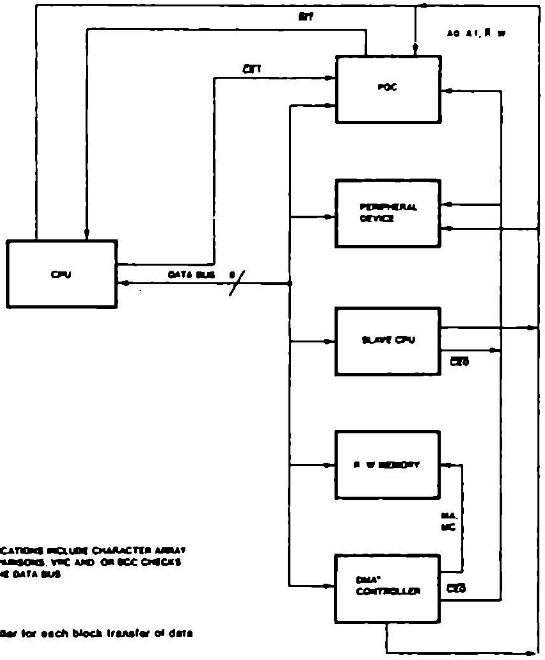

Rx HOLDING REGISTER READY
CHANNEL j
IS CHANNEL POINTER = j ?
NO
READ $BCC_{u,j}$ AND STORE IN BCC SAVE AREA SPECIFIED BY CHANNEL POINTER
READ MODE REGISTER AND STORE IN SAVE AREA.
CHANNEL j BCC AND MODE DATA $BCC_{u,j}$ AND MODE REGISTER
j - CHANNEL POINTER
YES
READ CHARACTER FROM R/T AND LOAD INTO PGC VIA $\overline{CEB}$
READ PGC STATUS REGISTER
CHECK STATUS BITS AND PROCEED AS REQUIRED

Figure 12

PGC DATA BUS MONITORING WITH DMA TRANSFERS



APPLICATIONS INCLUDE CHARACTER ARRAY COMPARISONS, VRC AND/OR ECC CHECKS ON THE DATA BUS

NOTE

*CPU initializes DMA controller for each block transfer of data

Figure 13

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	-65 to +150	°C
All voltages with respect to ground[3]	-0.5 to +6.0	V

NOTES

1. Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation sections of this specification is not implied.
2. For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (ceramic package) or 137°C/W (plastic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. However, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

POLYNOMIAL GENERATOR CHECKER (PGC)

SC2653

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ± 5%

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 2.2mA		0.25	0.45	
V_{OH}	High	I_{OH} = −400μA	2.4	2.8		
I_{IL}	Input load current	V_{IN} = 0 to 5.5V			10	μA
	Output leakage current					μA
I_{LD}	Data bus	V_{OUT} = 4.0V			10	
I_{LO}	Open drain	V_{OUT} = 4.0V			10	
I_{CC}	Power supply current			45	75	mA

AC CHARACTERISTICS T_A = 0° to +70°C, V_{CC} = 5V ± 5%[1, 2, 3]

PARAMETER		LIMITS		UNIT
		Min	Max	
t_{CE}	Chip enable pulse width	250		ns
t_{CED}	Chip enable period D	1750		ns
t_{CEC}[4]	Chip enable period C	1750		ns
t_{AS}	Address setup	10		ns
t_{AH}	Address hold	10		ns
t_{CS}	Control setup	10		ns
t_{CH}	Control hold	10		ns
t_{DS}[5]	Data setup	150		ns
t_{DH}	Data hold	10		ns
t_{DD}[6]	Data delay time for read		200	ns
t_{DF}[6]	Data bus floating time for read		100	ns
t_{INTL}[7]	Interrupt low delay		1600	ns
t_{INTH}[7]	Interrupt high delay		600	ns

NOTES

1. Parameters are valid over operating temperature range unless otherwise specified
2. All voltage measurements are referenced to ground. All time measurements are at 50% level for inputs and at the 0.8V or 2.0V level for outputs. Input levels for testing are 0.45V and 2.4V
3. Typical values are at +25°C, typical supply voltages and typical processing parameters
4. t_{CEC} = 800ns during PGC initialization when no BCC accumulation is in progress
5. t_{DS} = 60ns whenever $\overline{CED}$ is used
6. Test conditions: C_L = 150 pF
7. $\overline{INT}$ is an open drain output

PGC TIMING

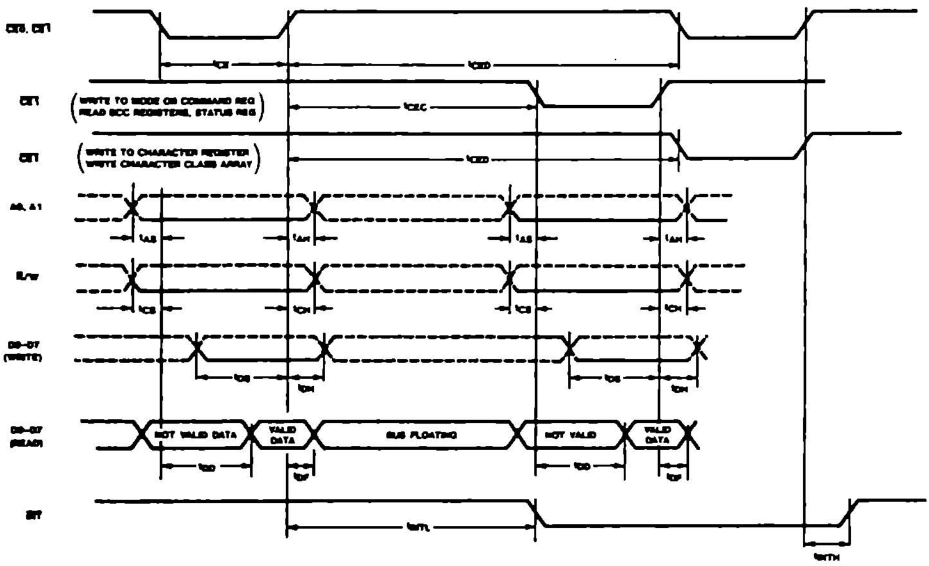

CE0, CE1
CE1
(WRITE TO MODE OR COMMAND REG
READ BCC REGISTERS, STATUS REG)
CE1
(WRITE TO CHARACTER REGISTER
WRITE CHARACTER CLASS ARRAY)
A0, A1
R/W
D0–D7
(WRITE)
D0–D7
(READ)
tCE
tCED
tCEC
tCED
tAS
tAH
tCS
tCH
tDS
tDH
NOT VALID DATA
VALID DATA
BUS FLOATING
NOT VALID
tDD
tDF

Figure 14

ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI) SC2661

DESCRIPTION

The Signetics 2661 EPCI is a universal synchronous/asynchronous data communications controller chip that is an enhanced pin compatible version of the 2651. It interfaces directly to most 8-bit microprocessors and may be used in a polled or interrupt driven system environment. The 2661 accepts programmed instructions from the microprocessor while supporting many serial data communications disciplines—synchronous and asynchronous—in the full or half-duplex mode. Special support for BISYNC is provided.

The EPCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The 2661 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode. Each version of the EPCI (A, B, C) has a different set of baud rates.

The EPCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

FEATURES

- Synchronous operation
  - 5 to 8-bit characters plus parity
  - Single or double SYN operation
  - Internal or external character synchronization
  - Transparent or non-transparent mode
  - Transparent mode DLE stuffing (Tx) and detection (Rx)
  - Automatic SYN or DLE-SYN insertion
  - SYN, DLE and DLE-SYN stripping
  - Odd, even, or no parity
  - Local or remote maintenance loop back mode
  - Baud rate: dc to 1M bps (1X clock)
- Asynchronous operation
  - 5 to 8-bit characters plus parity
  - 1, 1½ or 2 stop bits transmitted
  - Odd, even, or no parity
  - Parity, overrun and framing error detection
  - Line break detection and generation
  - False start bit detection
  - Automatic serial echo mode (echoplex)
  - Local or remote maintenance loop back mode
  - Baud rate: dc to 1M bps (1X clock)
    - dc to 62.5K bps (16X clock)
    - dc to 15.625K bps (64X clock)

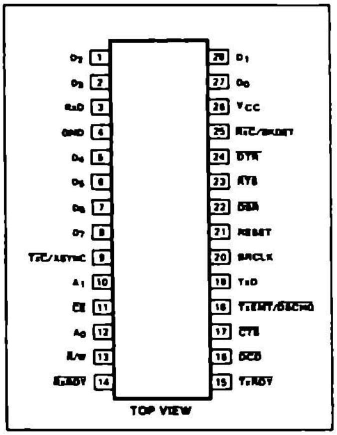
OTHER FEATURES

- Internal or external baud rate clock
- 3 baud rate sets
- 16 internal rates for each set
- Double buffered transmitter and receiver
- Dynamic character length switching
- Full or half duplex operation
- Fully compatible with 2650 CPU
- TTL compatible inputs and outputs
- RxC and TxC pins are short circuit protected
- 3 open drain MOS outputs can be wire-ORed
- Single 5V power supply
- No system clock required
- 28-pin dual in-line package

APPLICATIONS

- Intelligent terminals
- Network processors
- Front end processors
- Remote data concentrators
- Computer to computer links
- Serial peripherals
- BISYNC adaptors

PIN CONFIGURATION



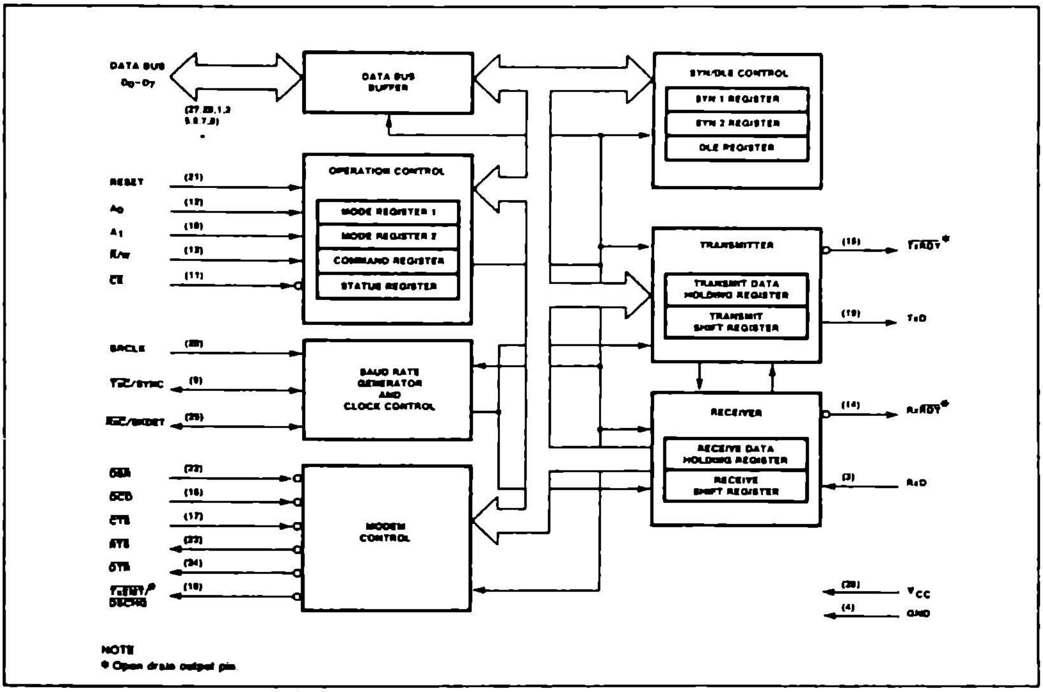
ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
Ceramic DIP	SC2661ACSI28 SC2661BCSI28 SC2661CCSI28	See table 1 for baud rates
Plastic DIP	SC2661ACSN28 SC2661BCSN28 SC2661CCSN28	See table 1 for baud rates

PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27,28,1,2,5-8	D_0-D_7	8-bit data bus	I/O
21	RESET	Reset	I
12,10	A_0-A_1	Internal register select lines	I
13	$\overline{R}$/W	Read or write command	I
11	$\overline{CE}$	Chip enable input	I
22	$\overline{DSR}$	Data set ready	I
24	$\overline{DTR}$	Data terminal ready	O
23	$\overline{RTS}$	Request to send	O
17	$\overline{CTS}$	Clear to send	I
16	$\overline{DCD}$	Data carrier detected	I
18	$\overline{TxEMT}/\overline{DSCHG}$	Transmitter empty or data set change	O
9	$\overline{TxC}$/XSYNC	Transmitter clock/external SYNC	I/O
25	$\overline{RxC}$/BKDET	Receiver clock/break detect	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	$\overline{TxRDY}$	Transmitter ready	O
14	$\overline{RxRDY}$	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	V_{CC}	+5V supply	I
4	GND	Ground	I

BLOCK DIAGRAM


DATA BUS
D_0-D_7
(27,28,1,2
5,6,7,8)
DATA BUS BUFFER
RESET
(21)
A_0
(12)
A_1
(10)
$R/\overline{W}$
(13)
$\overline{CE}$
(11)
OPERATION CONTROL
MODE REGISTER 1
MODE REGISTER 2
COMMAND REGISTER
STATUS REGISTER
BRCLK
(20)
$\overline{TxC}$/SYNC
(9)
$\overline{RxC}$/BKDET
(25)
BAUD RATE GENERATOR AND CLOCK CONTROL
$\overline{DSR}$
(22)
$\overline{DCD}$
(16)
$\overline{CTS}$
(17)
$\overline{RTS}$
(23)
$\overline{DTR}$
(24)
$\overline{TxEMT}$/$\overline{DSCHG}$*
(18)
MODEM CONTROL
SYN/DLE CONTROL
SYN 1 REGISTER
SYN 2 REGISTER
DLE REGISTER
TRANSMITTER
TRANSMIT DATA HOLDING REGISTER
TRANSMIT SHIFT REGISTER
(15)
$\overline{TxRDY}$*
(19)
TxD
RECEIVER
RECEIVE DATA HOLDING REGISTER
RECEIVE SHIFT REGISTER
(14)
$\overline{RxRDY}$*
(3)
RxD
(26)
V_{CC}
(4)
GND
NOTE
* Open drain output pin.

BLOCK DIAGRAM

The EPCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the EPCI programming section of this data sheet.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS SC2661A (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6144
0001	75	1.2	-	4096
0010	110	1.7598	−0.01	2793
0011	134.5	2.152	-	2284
0100	150	2.4	-	2048
0101	200	3.2	-	1536
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	-	256
1010	1800	28.7438	−0.19	171
1011	2000	31.9168	−0.26	154
1100	2400	38.4	-	128
1101	4800	76.8	-	64
1110	9600	153.6	-	32
1111	19200	307.2	-	16

ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI) SC2661

Timing

The EPCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See table 1.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

Table 1 BAUD RATE GENERATOR CHRACTERISTICS (Cont'd)

SC2661B (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	-	6144
0010	75	1.2	-	4096
0011	110	1.7598	−0.01	2793
0100	134.5	2.152	-	2284
0101	150	2.4	-	2048
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1200	19.2	-	256
1001	1800	28.7438	−0.19	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	-	128
1100	4800	76.8	-	64
1101	9600	153.6	-	32
1110	19200	307.2	-	16
1111	38400	614.4	-	8

SC2661C (BRCLK = 5.0688MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6336
0001	75	1.2	-	4224
0010	110	1.76	-	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	-	2112
0101	300	4.8	-	1056
0110	600	9.6	-	528
0111	1200	19.2	-	264
1000	1800	28.8	-	176
1001	2000	32.081	0.253	158
1010	2400	38.4	-	132
1011	3600	57.6	-	88
1100	4800	76.8	-	66
1101	7200	115.2	-	44
1110	9600	153.6	-	33
1111	19200	316.8	3.125	16

NOTE

16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X and BRG can be used only for TxC

Table 2 CPU-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
V_{CC}	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the 2661. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	10,12	I	Address lines used to select internal EPCI registers.
$\overline{\text{R}}$/W	13	I	Read command when low, write command when high.
$\overline{\text{CE}}$	11	I	Chip enable command. When low, indicates that control and data lines to the EPCI are valid and that the operation specified by the $\overline{\text{R}}$/W, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the three-state condition.
D_7-D_0	8,7,6,5, 2,1,28,17	I/O	8-bit, three-state data bus used to transfer commands, data and status between EPCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{\text{TxRDY}}$	15	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{\text{RxRDY}}$	14	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{\text{TxEMT}}$/ $\overline{\text{DSCHG}}$	18	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

OPERATION

The functional operation of the 2661 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the EPCI programming section of the data sheet.

After programming, the EPCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2661 is conditioned to receive data when the $\overline{\text{DCD}}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low (mark to space) transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data are then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{\text{RxRDY}}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{\text{RxC}}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (framing error), the receiver will interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (RxD is low for the entire character as well as the stop bit), only one character consisting of all zeros (with the FE status bit SR5 set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 25 can be programmed to be a break detect output by appropriate setting of MR27-MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 25 will go low. Refer to the break detection timing diagram.

Table 3 DEVICE-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
BRCLK	20	I	Clock input to the internal baud rate generator (see table 1). Not required if external receiver and transmitter clocks are used.
*$\overline{\text{RxC}}$/BKDET	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1X/16X clock or a break detect output pin.
*$\overline{\text{TxC}}$/XSYNC	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1X/16X clock output or an external jam synchronization input.
RxD	3	I	Serial data input to the receiver. "Mark" is high, "space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{\text{DSR}}$	22	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes if CR2 or CR0 = 1.
$\overline{\text{DCD}}$	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes if CR2 or CR0 = 1. If $\overline{\text{DCD}}$ goes high while receiving, the RxC is internally inhibited.
$\overline{\text{CTS}}$	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{\text{DTR}}$	24	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{\text{RTS}}$	23	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. If the transmit shift register is not empty when CR5 is reset (1 to 0), then $\overline{\text{RTS}}$ will go high one TxC time after the last serial bit is transmitted.

NOTE
*$\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ outputs have short circuit protection max. $C_L = 100pF$. Outputs become open circuited upon detection of a zero pulled high or a one pulled low.

When the EPCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN(CR2). In this mode, as data are shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the EPCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization.) When synchronization has been achieved, the EPCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{\text{RxRDY}}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the holding register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

External jam synchronization can be achieved via pin 9 by appropriate setting of MR27-MR24. When pin 9 is an XSYNC input, the internal SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled. Each positive going signal on XSYNC will cause the receiver to establish synchronization on the rising edge of the next RxC pulse. Character assembly will start with the RxD input at this edge. XSYNC may be lowered on the next rising edge of RxC. This external synchronization will cause the SYN DETECT status bit to be set until the status register is read. Refer to XSYNC timing diagram.

Transmitter

The EPCI is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The 2661 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data are transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the $\overline{TxEMT}/\overline{DSCHG}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

In the synchronous mode, when the 2661 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the EPCI unless the CPU fails to send a new character to the EPCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the EPCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in the THR.

EPCI PROGRAMMING

Prior to initiating data communications, the 2661 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The EPCI can be reconfigured at any time during program execution. A flowchart of the initialization process appears in figure 1.

The internal registers of the EPCI are accessed by applying specific signals to the $\overline{CE}$, $\overline{R}/W$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in table 4.

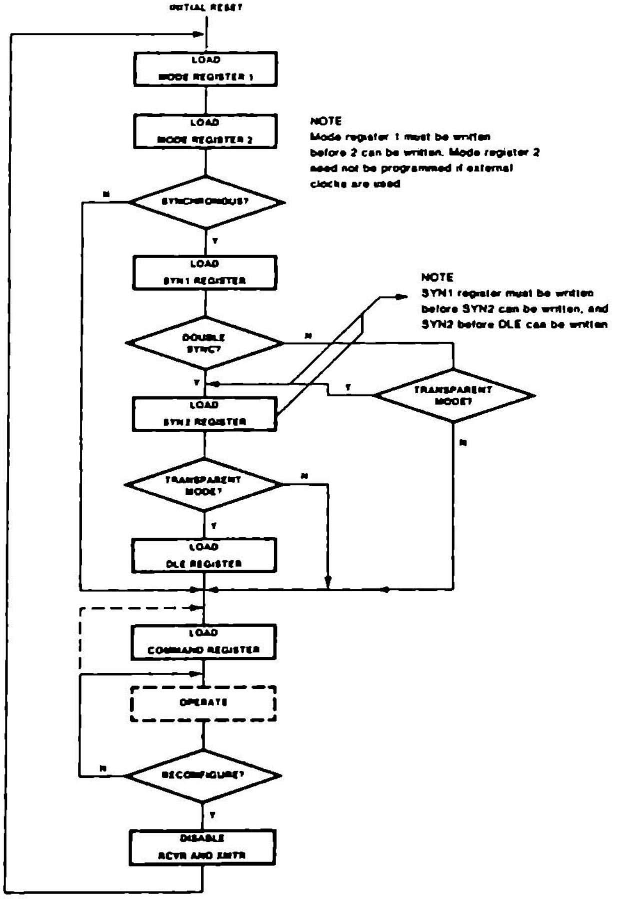
The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and

Table 4 2661 REGISTER ADDRESSING

$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
1	X	X	X	Three-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers ½
0	1	0	1	Write mode registers ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC characteristics section for timing requirements

2661 INITIALIZATION FLOW CHART



NOTE
Mode register 1 must be written before 2 can be written. Mode register 2 need not be programmed if external clocks are used

NOTE
SYN1 register must be written before SYN2 can be written, and SYN2 before DLE can be written

Figure 1

$\overline{R}$/W = 1. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a read command register operation, but are unaffected by any other read or write operation.

The 2661 register formats are summarized in tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the EPCI, while the command register controls the operation within this basic framework. The EPCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit.) In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used to establish character sync. When transmitting, a DLE character in the transmit holding register will cause a second DLE character to be transmitted. This DLE stuffing eliminates the software DLE compare and stuff on each transparent mode data character. If the send DLE command (CR3) is active when a DLE is loaded into THR, only one additional DLE will be transmitted. Also, DLE stripping and DLE detect (with MR14 = 0) are enabled.

The bits in the mode register affecting character assembly and disassembly (MR12-MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore in synchronous mode, changes should be made only in half duplex mode (RxEN = 1 or TxEN = 1, but not both simultaneously = 1). In asynchronous mode, character changes should be made when RxEN and TxEN=0 or when TxEN = 1 and the transmitter is marking in half duplex mode (RxEN = 0).

To effect assembly/disassembly of the next received/transmitted character, MR12-15 must be changed within n bit times of the active going state of $\overline{RxRDY}$/$\overline{TxRDY}$. Transparent and non-transparent mode changes (MR16) must occur within n-1 bit times of the character to be affected when the receiver or transmitter is active. (n = smaller of the new and old character lengths.)

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable for each EPCI version (−1, −2, −3). Version 1 and 2 specify a 4.9152 MHz TTL input at BRCLK (pin 20); version 3 specifies a 5.0688 MHz input which is identical to the Signetics 2651. MR23-20 are don't cares if external clocks are selected (MR25-MR24 = 0). The individual rates are given in table 1.

MR24-MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at pins 9 and 25. Refer to table 6.

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{RxC}$ rising edge. Disabling the receiver causes $\overline{RxRDY}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state

Table 5 MODE REGISTER 1 (MR 1)

MR17	MR16	MR15	MR14	MR13 MR12	MR11 MR10
Sync/Async		Parity Type	Parity Control	Character Length	Mode and Baud Rate Factor
Async: Stop Bit Length 00 = Invalid 01 = 1 stop bit 10 = 1½ stop bits 11 = 2 stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate
Sync: Number of SYN char 0 = Double SYN 1 = Single SYN	Sync: Transparency Control 0 = Normal 1 = Transparent				

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case

Table 6 MODE REGISTER 2 (MR2)

MR27–MR24											MR23–MR20
	TxC	RxC	Pin 9	Pin 25		TxC	RxC	Pin 9	Pin 25	Mode	Baud Rate Selection
0000	E	E	TxC	RxC	1000	E	E	XSYNC¹	RxC/TxC	sync	See baud rates in table 1
0001	E	I	TxC	1X	1001	E	I	TxC	BKDET	async	
0010	I	E	1X	RxC	1010	I	E	XSYNC¹	RxC	sync	
0011	I	I	1X	1X	1011	I	I	1X	BKDET	async	
0100	E	E	TxC	RxC	1100	E	E	XSYNC¹	RxC/TxC	sync	
0101	E	I	TxC	16X	1101	E	I	TxC	BKDET	async	
0110	I	E	16X	RxC	1110	I	E	XSYNC¹	RxC	sync	
0111	I	I	16X	16X	1111	I	I	16X	BKDET	async	

NOTES

1 When pin 9 is programmed as XSYNC input, SYN1, SYN1·SYN2, and DLE·SYN1 detection is disabled.

E = External clock

I = Internal clock (BRG)

1X and 16X are clock outputs

Table 7 COMMAND REGISTER (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	Request To Send	Reset Error	Sync/Async	Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = Normal operation 01 = Async: Automatic echo mode Sync: SYN and/or DLE stripping mode 10 = Local loop back 11 = Remote loop back	0 = Force $\overline{\text{RTS}}$ output high one clock time after TxSR serialization 1 = Force $\overline{\text{RTS}}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE/DLE detect)	Async: Force break 0 = Normal 1 = Force break Sync: Send DLE 0 = Normal 1 = Send DLE	0 = Disable 1 = Enable	0 = Force $\overline{\text{DTR}}$ output high 1 = Force $\overline{\text{DTR}}$ output low	0 = Disable 1 = Enable

Table 8 STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{\text{DSR}}$ input is high 1 = $\overline{\text{DSR}}$ input is low	0 = $\overline{\text{DCD}}$ input is high 1 = $\overline{\text{DCD}}$ input is low	Async: 0 = Normal 1 = Framing Error Sync: 0 = Normal 1 = SYN detected	0 = Normal 1 = Overrun Error	Async: 0 = Normal 1 = Parity error Sync: 0 = Normal 1 = Parity error or DLE received	0 = Normal 1 = Change in $\overline{\text{DSR}}$, or $\overline{\text{DCD}}$, or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty

(high) while $\overline{\text{TxRDY}}$ and $\overline{\text{TxEMT}}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected. A 0 to 1 transition of CR2 will initiate start bit search (async) or hunt mode (sync).

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ outputs. Data at the outputs are the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. Since this is a one time command, CR3 does not have to be reset by software. CR3 should be set when entering and exiting transparent mode and for all DLE—non-DLE character sequences.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

Table 9 SC2661 EPCI vs SC2651 PCI

FEATURE	EPCI	PCI
1. MR2 Bit 6, 7	Control pin 9, 25	Not used
2. DLE detect-SR3	SR3 = 0 for DLE-DLE, DLE-SYNC1	SR3 = 1 for DLE-DLE, DLE-SYNC1
3. Reset of SR3, DLE detect	Second character after DLE, or receiver disable, or CR4 = 1	Receiver disable, or CR4 = 1
4. Send DLE-CR3	One time command	Reset via CR3 on next $\overline{\text{TxRDY}}$
5. DLE stuffing in transparent mode	Automatic DLE stuffing when DLE is loaded except if CR3 = 1	None
6. SYNC1 stripping in double sync non-transparent mode	All SYNC1	First SYNC1 of pair
7 Baud rate versions	Three	One
8 Terminate ASYNC transmission (drop RTS)	Reset CR5 in response to $\overline{\text{TxRDY}}$ changing from 1 to 0	Reset CR0 when $\overline{\text{TxEMT}}$ goes from 1 to 0. Then reset CR5 when $\overline{\text{TxEMT}}$ goes from 0 to 1
9. Break detect	Pin 25[1]	FE and null character
10 Stop bit searched	One	Two
11. External jam sync	Pin 9[2]	No
12. Data bus timing	Improved over 2651	—
13 Data bus drivers	Sink 2.2mA Source 400μA	Sink 1.6mA Source 100μA

NOTES

1 Internal BRG used for RxC

2 Internal BRG used for TxC

When CR5 (RTS) is set, the $\overline{\text{RTS}}$ pin is forced low and the transmit serial logic is enabled. A 1 to 0 transition of CR5 will cause $\overline{\text{RTS}}$ to go high (inactive) one TxC time after the last serial bit has been transmitted (if the transmit shift register was not empty).

The EPCI can operate in one of four sub-modes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7-CR6 = 01 places the EPCI in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{\text{TxRDY}}$ output = 1.
4. The $\overline{\text{TxEMT/DSCHG}}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7-CR6 = 01 places the EPCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In local loop back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{\text{DTR}}$ is connected to $\overline{\text{DCD}}$ and $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{\text{DTR}}$, $\overline{\text{RTS}}$ and $\overline{\text{TxD}}$ outputs are held high.
5. The $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, $\overline{\text{DSR}}$ and RxD inputs are ignored.

Additional requirements to operate in the local loop back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the EPCI.

The second diagnostic mode is the remote loop back mode (CR7-CR6 = 11). In this mode.

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output
2. The transmitter is clocked by the receive clock.
3. No data are sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{\text{RxRDY}}$, $\overline{\text{TxRDY}}$, and $\overline{\text{TxEMT/DSCHG}}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally

Status Register

The data contained in the status register (as shown in table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{\text{TxRDY}}$ output pin is low. In the automatic echo and remote loop back modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\text{RxRDY}}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$ inputs (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching DLE register was received and the present character is neither SYN1 nor DLE. This bit is cleared when the next character following the above sequence is loaded into RHR, when the receiver is disabled, or by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled or by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by a stop bit, i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, or when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{\text{DCD}}$ and $\overline{\text{DSR}}$ inputs respectively. A low input sets its corresponding status bit, and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5% [4,5,6]

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 2.2mA			0.4	
V_{OH}[7]	High	I_{OH} = −400μA	2.4			
I_{IL}	Input leakage current	V_{IN} = 0 to 5.5 V			10	μA
	3-state output leakage current					μA
I_{LH}	Data bus high	V_O = 4.0V			10	
I_{LL}	Data bus low	V_O = 0.45V			10	
I_{CC}	Power supply current				150	mA

CAPACITANCE T_A = 25°C, V_{CC} = 0V

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Capacitance					pF
C_{IN}	Input				20	
C_{OUT}	Output	fc = 1MHz			20	
$C_{I/O}$	Input/Output	Unmeasured pins tied to ground			20	

Notes on following page.

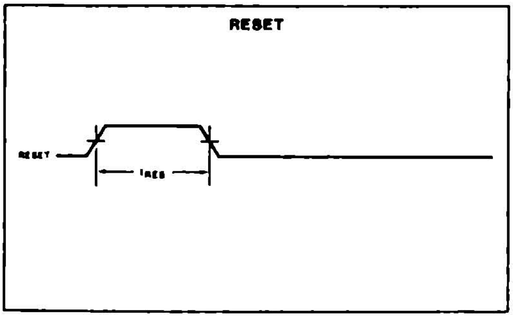
ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI) SC2661

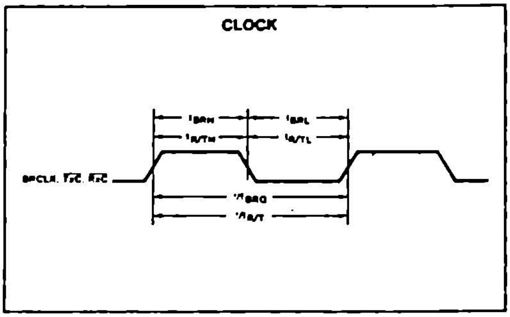
AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6]

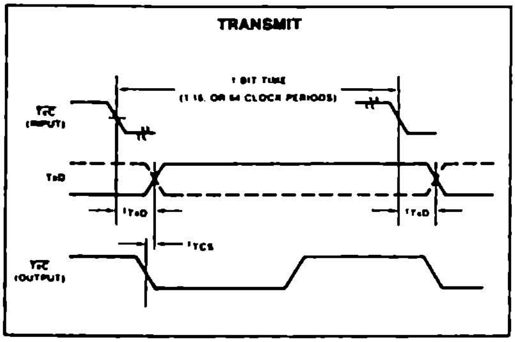
	PARAMETER	TEST CONDITIONS	Min	Typ	Max	UNIT
	Pulse width					ns
t_{RES}	Reset		1000			
t_{CE}	Chip enable		250			
	Setup and hold time					ns
t_{AS}	Address setup		10			
t_{AH}	Address hold		10			
t_{CS}	$\overline{R}$/W control setup		10			
t_{CH}	$\overline{R}$/W control hold		10			
t_{DS}	Data setup for write		150			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	$C_L = 150pF$			200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$			100	
t_{CED}	CE to CE delay		600			
	Input clock frequency					MHz
f_{BRG}	Baud rate generator (2661A,B)		1.0	4.9152	4.9202	
f_{BRG}	Baud rate generator (2661C)		1.0	5.0688	5.0738	
$f_{R/T}$[10]	$\overline{TxC}$ or $\overline{RxC}$		dc		1.0	
	Clock width					ns
t_{BRH}[9]	Baud rate high (2661A,B)		75			
t_{BRH}[9]	Baud rate high (2661C)		70			
t_{BRL}[9]	Baud rate low (2661A,B)		75			
t_{BRL}[9]	Baud rate low (2661C)		70			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		480			
$t_{R/TL}$[10]	$\overline{TxC}$ or $\overline{RxC}$ low		480			
t_{TXD}	TxD delay from falling edge of $\overline{TxC}$	$C_L = 150pF$			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output[8]	$C_L = 150pF$		0		

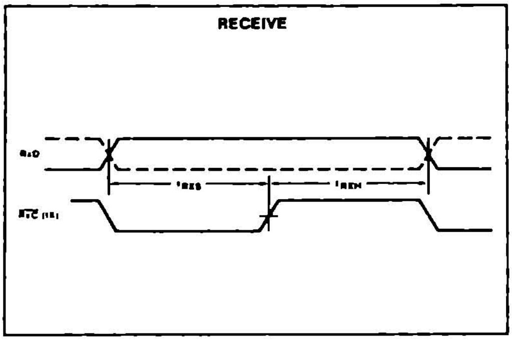
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (IO ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8V and 2.0V for outputs. Input levels swing between 0.4V and 2.4V, with a transition time of 20 ns maximum.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. $\overline{TxRDY}$, $\overline{RxRDY}$ and $\overline{TxEMT}/\overline{DSCHG}$ outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688 MHz f_{BRG} (2661C) and 4.9152 MHz f_{BRG} (2661A,B), t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.
10. In asynchronous local loopback mode, using 1X clock, the following parameters apply:
    $f_{R/T}$ = 0.83 MHz max.
    $t_{R/TL}$ = 700 ns min.

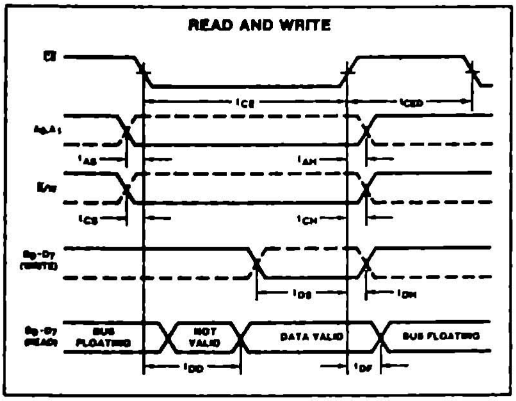
TIMING DIAGRAMS


RESET
RESET
tRES


CLOCK
tBRH
tBRL
tR/TH
tR/TL
BRCLK, TxC, RxC
1/fBRG
1/fR/T

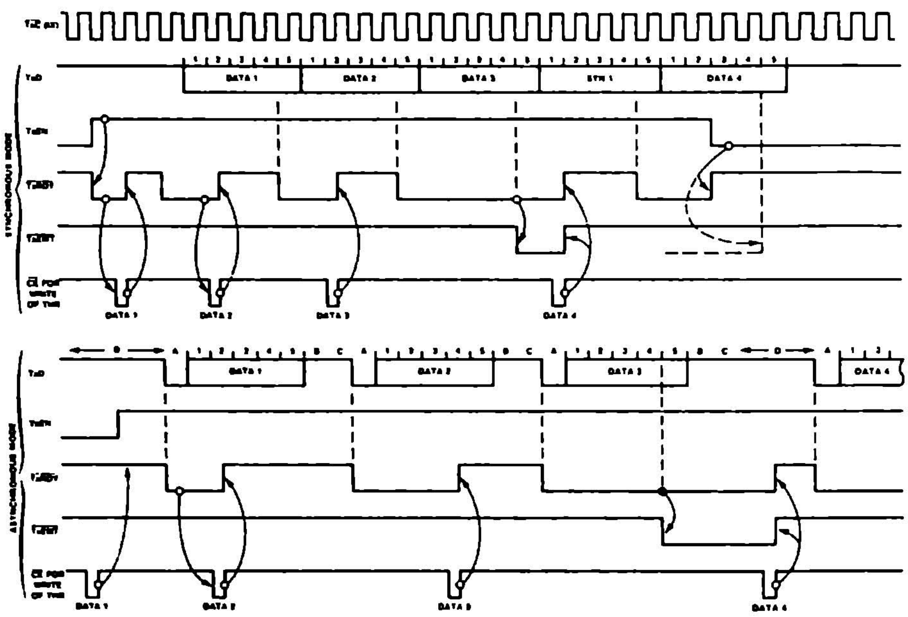

TRANSMIT
1 BIT TIME
(1, 16, OR 64 CLOCK PERIODS)
TxC (INPUT)
TxD
tTxD
tTxD
tTCS
TxC (OUTPUT)


RECEIVE
RxD
tRXS
tRXH
RxC (1X)


READ AND WRITE
CE
tCE
tCED
A0,A1
tAS
tAH
R/W
tCS
tCH
D0-D7 (WRITE)
tDS
tDH
D0-D7 (READ)
BUS FLOATING
NOT VALID
DATA VALID
BUS FLOATING
tDD
tDF

TIMING DIAGRAMS (Cont'd)

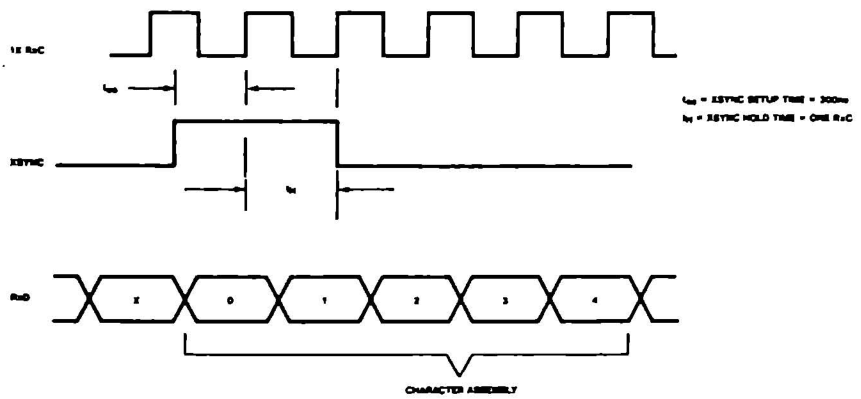
TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])


TxC (1X)
SYNCHRONOUS MODE
TxD
DATA 1
DATA 2
DATA 3
SYN 1
DATA 4
TxEN
TxRDY
TxEMT
CE FOR WRITE OF THR
DATA 1
DATA 2
DATA 3
DATA 4
ASYNCHRONOUS MODE
TxD
A
B
C
D
DATA 1
DATA 2
DATA 3
DATA 4
TxEN
TxRDY
TxEMT
CE FOR WRITE OF THR
DATA 1
DATA 2
DATA 3
DATA 4

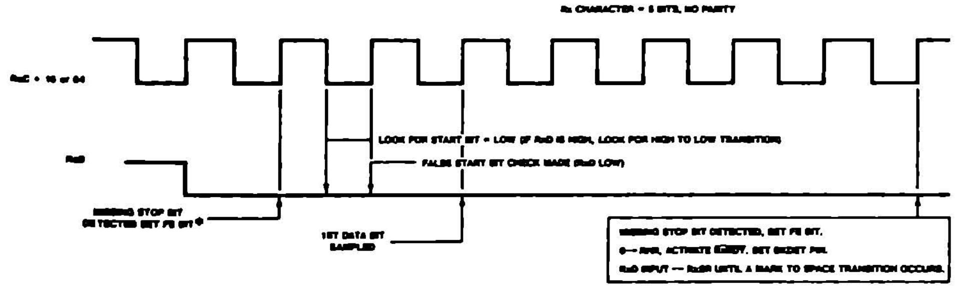
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit

TIMING DIAGRAMS (Cont'd)

EXTERNAL SYNCHRONIZATION WITH XSYNC


1X RxC
tXS
tXS = XSYNC SETUP TIME = 300ns
tXH = XSYNC HOLD TIME = ONE RxC
XSYNC
tXH
RxD
X
0
1
2
3
4
CHARACTER ASSEMBLY

BREAK DETECTION TIMING

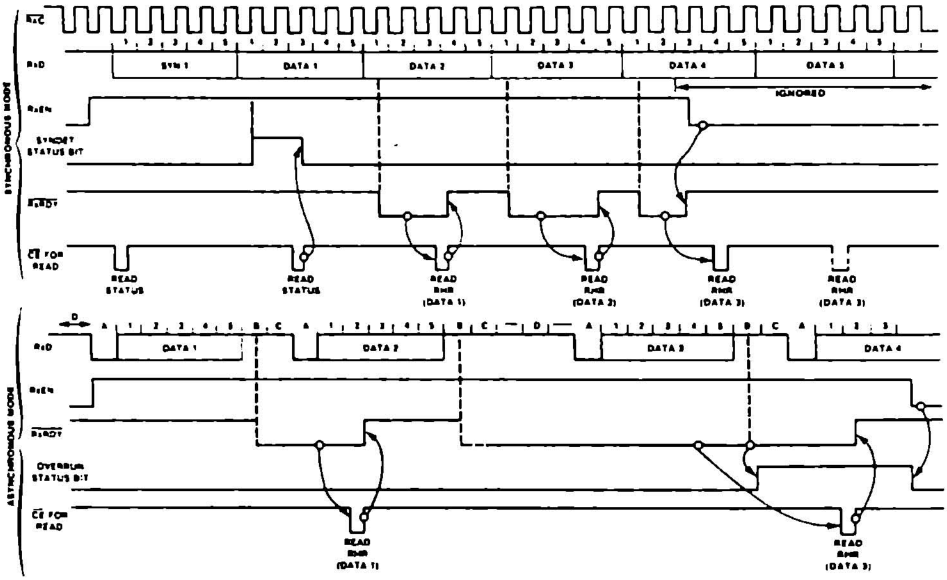

Rx CHARACTER = 5 BITS, NO PARITY
RxC = 16 or 64
LOOK FOR START BIT = LOW (IF RxD IS HIGH, LOOK FOR HIGH TO LOW TRANSITION)
FALSE START BIT CHECK MADE (RxD LOW)
RxD
MISSING STOP BIT
DETECTED SET FE BIT◆
1ST DATA BIT
SAMPLED
MISSING STOP BIT DETECTED, SET FE BIT.
0 → RHR, ACTIVATE RxRDY, SET BKDET BIT.
RxD INPUT — RxSR UNTIL A MARK TO SPACE TRANSITION OCCURS.

NOTE

◆ If the stop bit is present, the start bit search will commence immediately.

TIMING DIAGRAMS (Cont'd)

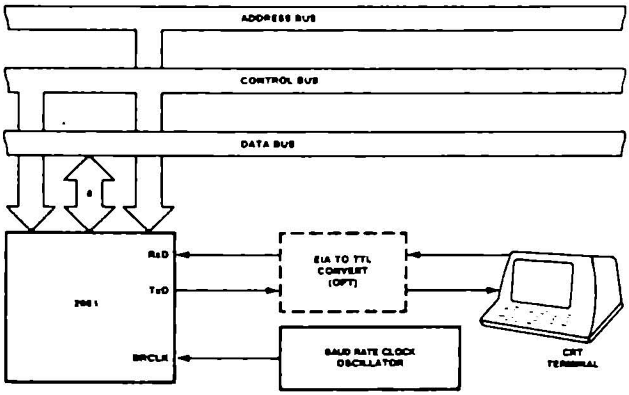
RxRDY (Shown for 5-bit characters, no parity, 2 stops bits [in asynchronous mode])


SYNCHRONOUS MODE
RxC
RxD
SYN 1
DATA 1
DATA 2
DATA 3
DATA 4
DATA 5
RxEN
IGNORED
SYNDET STATUS BIT
RxRDY
CE FOR READ
READ STATUS
READ STATUS
READ RHR (DATA 1)
READ RHR (DATA 2)
READ RHR (DATA 3)
READ RHR (DATA 3)
ASYNCHRONOUS MODE
D
A
B
C
RxD
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
RxRDY
OVERRUN STATUS BIT
CE FOR READ
READ RHR (DATA 1)
READ RHR (DATA 3)

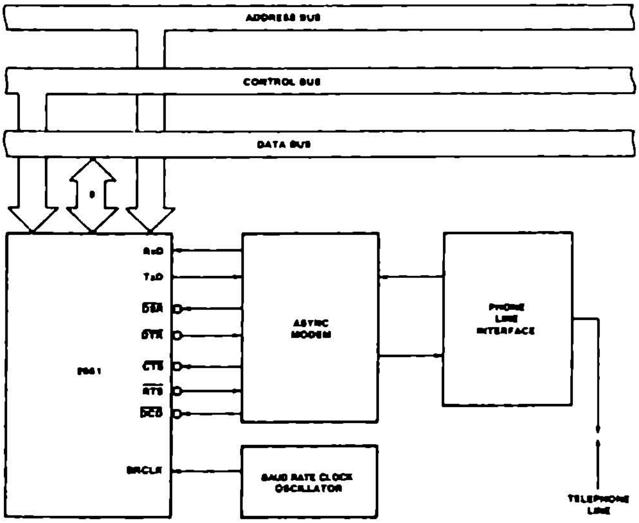
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
Only one stop bit is detected

TYPICAL APPLICATIONS

ASYNCHRONOUS INTERFACE TO CRT TERMINAL

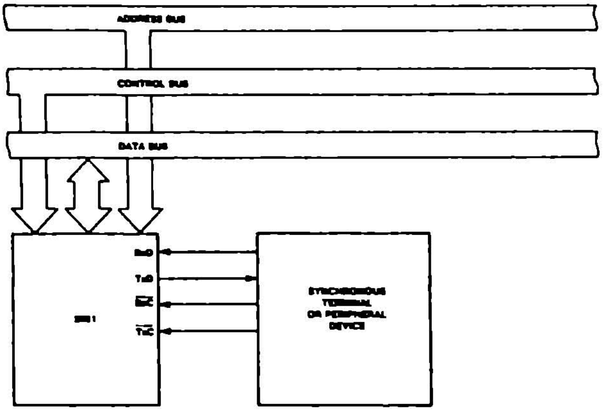

ADDRESS BUS
CONTROL BUS
DATA BUS
8
2661
RxD
TxD
BRCLK
EIA TO TTL
CONVERT
(OPT)
BAUD RATE CLOCK
OSCILLATOR
CRT
TERMINAL

ASYNCHRONOUS INTERFACE TO TELEPHONE LINES

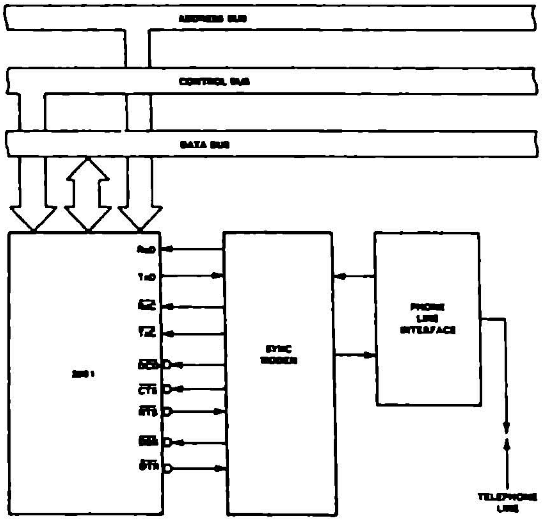

ADDRESS BUS
CONTROL BUS
DATA BUS
8
2661
RxD
TxD
DSR
DTR
CTS
RTS
DCD
BRCLK
ASYNC
MODEM
PHONE
LINE
INTERFACE
BAUD RATE CLOCK
OSCILLATOR
TELEPHONE
LINE

TYPICAL APPLICATIONS (Cont'd)

SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE


ADDRESS BUS
CONTROL BUS
DATA BUS
RxD
TxD
RxC
TxC
2661
SYNCHRONOUS
TERMINAL
OR PERIPHERAL
DEVICE

SYNCHRONOUS INTERFACE TO TELEPHONE LINES


ADDRESS BUS
CONTROL BUS
DATA BUS
RxD
TxD
RxC
TxC
DCD
CTS
RTS
DSR
DTR
2661
SYNC
MODEM
PHONE
LINE
INTERFACE
TELEPHONE
LINE

JANUARY 1982

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC2681 SERIES

Preview

DESCRIPTION

The Signetics SC2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

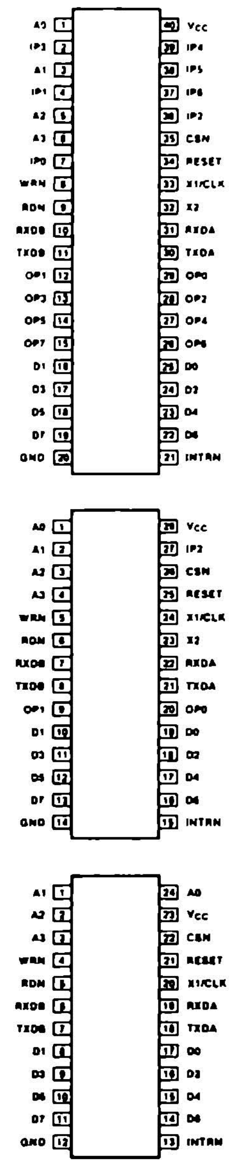
Also provided on the SC2681 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SC2681 is available in three package versions to satisfy various system requirements: 40-pin and 28-pin, both 0.6" wide DIPs, and a compact 24-pin, 0.4" wide, DIP.

FEATURES

- Dual full-duplex asynchronous receiver/transmitter
- Quadruple buffered receiver data registers
- Programmable data format
  - —5 to 8 data bits plus parity
  - —Odd, even, no parity or force parity
  - —1, 1.5 or 2 stop bits programmable in 1/16 bit increments
- Programmable baud rate for each receiver and transmitter selectable from:
  - —18 fixed rates: 50 to 38.4K baud
  - —One user defined rate derived from programmable timer/counter
  - —External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - —Normal (full duplex)
  - —Automatic echo
  - —Local loopback
  - —Remote loopback
- Multi-function programmable 16-bit counter/timer
- Multi-function 7-bit input port
  - —Can serve as clock or control inputs
  - —Change of state detection on four inputs
- Multi-function 8-bit output port
  - —Individual bit set/reset capability
  - —Outputs can be programmed to be status/interrupt signals
- Versatile interrupt system
  - —Single interrupt output with eight maskable interrupting conditions
  - —Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs
- Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec
- Automatic wake-up mode for multidrop applications
- Start-end break interrupt/status
- Detects break which originates in the middle of a character
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply

PIN CONFIGURATION


A0 1
IP3 2
A1 3
IP1 4
A2 5
A3 6
IP0 7
WRN 8
RDN 9
RXDB 10
TXDB 11
OP1 12
OP3 13
OP5 14
OP7 15
D1 16
D3 17
D5 18
D7 19
GND 20
40 VCC
39 IP4
38 IP5
37 IP6
36 IP2
35 CSN
34 RESET
33 X1/CLK
32 X2
31 RXDA
30 TXDA
29 OP0
28 OP2
27 OP4
26 OP6
25 D0
24 D2
23 D4
22 D6
21 INTRN
A0 1
A1 2
A2 3
A3 4
WRN 5
RDN 6
RXDB 7
TXDB 8
OP1 9
D1 10
D3 11
D5 12
D7 13
GND 14
28 VCC
27 IP2
26 CSN
25 RESET
24 X1/CLK
23 X2
22 RXDA
21 TXDA
20 OP0
19 D0
18 D2
17 D4
16 D6
15 INTRN
A1 1
A2 2
A3 3
WRN 4
RDN 5
RXDB 6
TXDB 7
D1 8
D3 9
D5 10
D7 11
GND 12
24 A0
23 VCC
22 CSN
21 RESET
20 X1/CLK
19 RXDA
18 TXDA
17 D0
16 D2
15 D4
14 D6
13 INTRN

TOP VIEWS

ORDERING CODE

PACKAGES	$V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $70°C$		
	24 Pin[1]	28 Pin[2]	40 Pin[2]
Ceramic DIP Plastic DIP	SC2681CSI24 SC2681CSN24	SC2681CSI28 SC2681CSN28	SC2681CSI40 SC2681CSN40

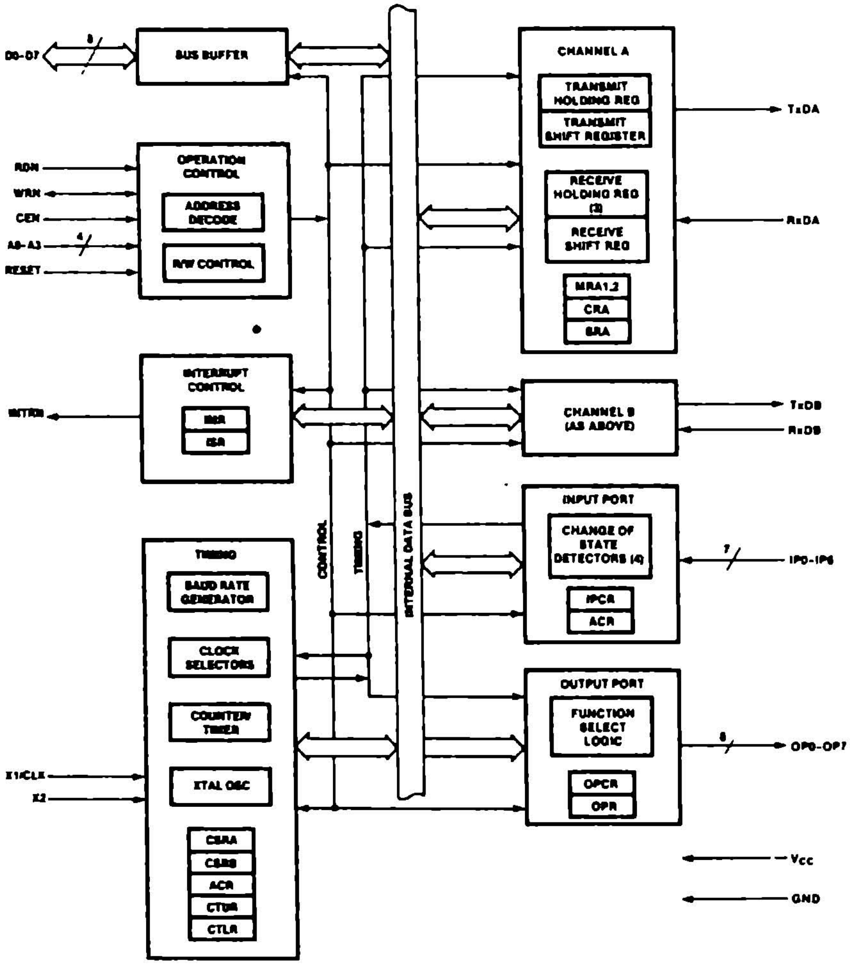
[1] 400 mil wide DIP

[2] 600 mil wide DIP

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC2681 SERIES

Preview

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
RDN
WRN
CEN
A0-A3
4
RESET
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTRN
INTERRUPT CONTROL
IMR
ISR
X1/CLK
X2
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/TIMER
XTAL OSC
CSRA
CSRB
ACR
CTUR
CTLR
CONTROL
TIMING
INTERNAL DATA BUS
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
MRA1,2
CRA
SRA
TxDA
RxDA
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
CHANGE OF STATE DETECTORS (4)
IPCR
ACR
7
IP0-IP6
OUTPUT PORT
FUNCTION SELECT LOGIC
OPCR
OPR
8
OP0-OP7
Vcc
GND

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC2681 SERIES

Preview

PIN DESIGNATION

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0-D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the WRN, RDN and A0-A3 inputs. When high, places the D0-D7 lines in the 3-state condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0-A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0-OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times.
X2	X	X		I	**Crystal 2:** Connection for other side of the crystal. Should be open if crystal is not used.
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X.	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated on receive or transmit.
OP2	X	X		O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input.
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.

PIN DESIGNATION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
IP4	X			I	Input 4: General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	Input 5: General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	Input 6: General purpose input or channel B receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	Power Supply: +5V supply input
GND	X	X	X	I	Ground

BLOCK DIAGRAM

The 2681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3–OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the 2681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25-50µs will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 2681 is conditioned to transmit data when the transmitter is enabled through the command register. The 2681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. if it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The 2681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the

Preview

receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4·3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed slave' station. The slave stations, with eceivers that are normally disabled, exmine the received data stream and 'wakeip' the CPU (by setting RxRDY) only upon eceipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the ac-

Table 1. 2681 REGISTER ADDRESSING

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	'Reserved'	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	'Reserved'	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	'Reserved'	'Reserved'
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC2681 SERIES

Preview

Table 2. REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A MR1B	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = special mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0–7 if channel is programmed for 5 bits/char

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
CSRA CSRB	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	not used—must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
CRA CRB		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/ FFULLB	0 = OPR[4] 1 = RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

Table 2. REGISTER BIT FORMATS (continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

cumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.

Preview

3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate	
CSRA[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4K	19.2K
1 1 0 1	Timer	Timer
1 1 1 0	IP4—16X	IP4—16X
1 1 1 1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP3—16X	IP3—16X
1 1 1 1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register — Access Type: Write Only

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP6—16X	IP6—16X
1 1 1 1	IP6—1X	IP6—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP5—16X	IP5—16X
1 1 1 1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

Preview

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the following:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

Table 3. BAUD RATE GENERATOR CHARACTERISTICS
CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE
Duty cycle of 16X clock is 50% ± 1%

Preview

ACR[6:4]—Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

Table 4. ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

ACR[3:0] — IP3, IP2, IP1, IP0 Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change in Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR2B[5]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR2A[5]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3–OP7 or the reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSB's and eight LSB's respectively of the value to be used by the counter/timer in either the counter or timer modes of operation.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1111) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR. The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1110). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to + 70	°C
Storage temperature	− 65 to + 150	°C
All voltages with respect to ground[3]	− 0.5 to + 6.0	V

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on + 150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient for ceramic package (116°C/W for plastic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%^{4,5,6}$

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		3.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	− 10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	− 10		10	μA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	− 10		10	μA
I_{CC}	Power supply current				150	mA

NOTES:

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at + 25°C, typical supply voltages, and typical processing parameters.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC2681 SERIES

Preview

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6,7]

PARAMETER	TENTATIVE LIMITS			UNIT
	Min	Typ	Max	
Reset Timing (figure 1)				
t_{RES} RESET pulse width	1.0			µs
Bus Timing (figure 2)[8]				
t_{AS} A0–A3 setup time to RDN, WRN low	10			ns
t_{AH} A0–A3 hold time from RDN, WRN high	10			ns
t_{CS} CEN setup time to RDN, WRN low	0			ns
t_{CH} CEN hold time from RDN, WRN high	0			ns
t_{RW} WRN, RDN pulse width	225			ns
t_{DD} Data valid after RDN low			150	ns
t_{DF} Data bus floating after RDN high			100	ns
t_{DS} Data setup time before WRN high	100			ns
t_{DH} Data hold time after WRN high	0			ns
t_{CC} Time between READs and/or WRITEs	600			ns
Port Timing (figure 3)[8]				
t_{PS} Port input setup time before RDN low	0			ns
t_{PH} Port input hold time after RDN high	0			ns
t_{PD} Port output valid after WRN high			400	ns
Interrupt Timing (figure 4)				
t_{IR} INTRN (or OP3–OP7 when used as interrupts) high from:				
Read RHR (RXRDY/FFULL interrupt)			300	ns
Write THR (TXRDY interrupt)			300	ns
Reset command (delta break interrupt)			300	ns
Stop C/T command (counter interrupt)			300	ns
Read IPCR (input port change interrupt)			300	ns
Write IMR (clear of interrupt mask bit)			200	ns
Clock Timing (figure 5)				
t_{CLK} X1/CLK high or low time	100			ns
f_{CLK} X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC} CTCLK (IP2) high or low time	200			ns
f_{CTC} CTCLK (IP2) frequency	0		2.0	MHz
t_{RX} RxC high or low time	220			ns
f_{RX} RxC frequency (16X)	0		2.0	MHz
(1X)	0		1.0	MHz
t_{TX} TxC high or low time	220			ns
f_{TX} TxC frequency (16X)	0		2.0	MHz
(1X)	0		1.0	MHz
Transmitter Timing (figure 6)				
t_{TXD} TxD output delay from TxC low			350	ns
t_{TCS} TxC output skew from TxD output data	−75	0	75	ns
Receiver Timing (figure 7)				
t_{RXS} RxD data setup time to RXC high	200			ns
t_{RXH} RxD data hold time from RXC high	200			ns

NOTES

[4] Parameters are valid over specified temperature range

[5] All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.

[6] Typical values are at +25°C, typical supply voltages, and typical processing parameters

[7] Test condition for outputs $C_L = 150pF$

[8] Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN

Preview

RESET TIMING

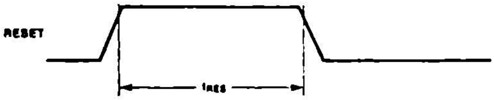


FIGURE 1

BUS TIMING

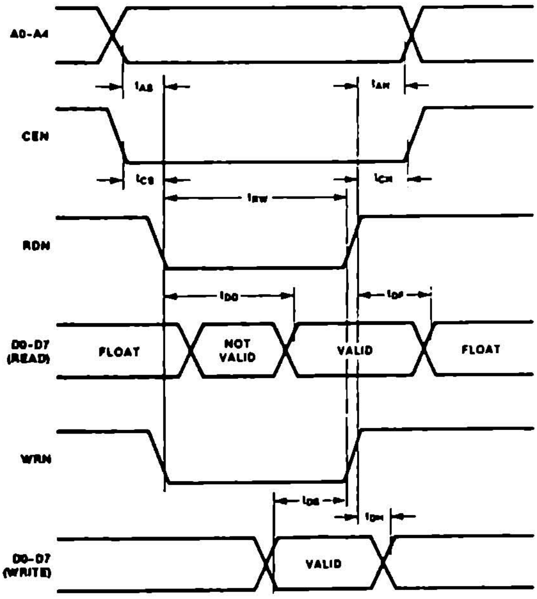

A0-A4
t_{AS}
t_{AH}
CEN
t_{CS}
t_{CH}
t_{RW}
RDN
t_{DD}
t_{DF}
D0-D7 (READ)
FLOAT
NOT VALID
VALID
FLOAT
WRN
t_{DS}
t_{DH}
D0-D7 (WRITE)
VALID

FIGURE 2

Preview

PORT TIMING

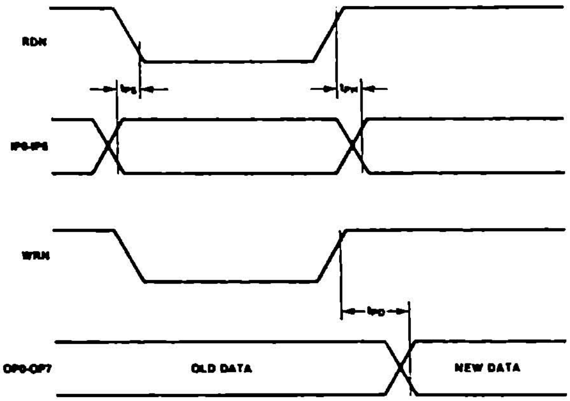

RDN
tPS
tPH
IP0-IP6
WRN
tPD
OP0-OP7
OLD DATA
NEW DATA

FIGURE 3

INTERRUPT TIMING

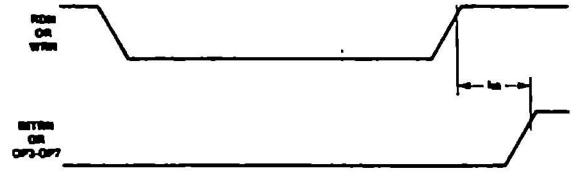

RDN
OR
WRN
tIR
INTRN
OR
OP3-OP7

FIGURE 4

CLOCK TIMING

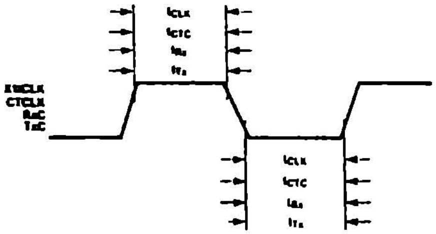

X1/CLK
CTCLK
RxC
TxC
tCLK
tCTC
tRX
tTX
tCLK
tCTC
tRX
tTX

FIGURE 5

TRANSMIT

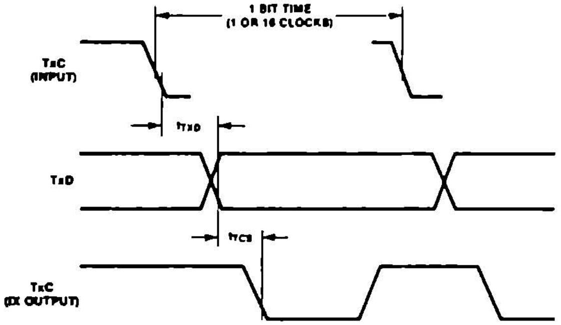

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
tTXD
TxD
tTCS
TxC
(1X OUTPUT)

FIGURE 6

RECEIVE

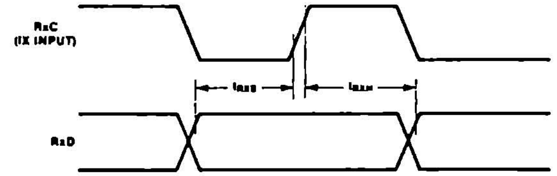
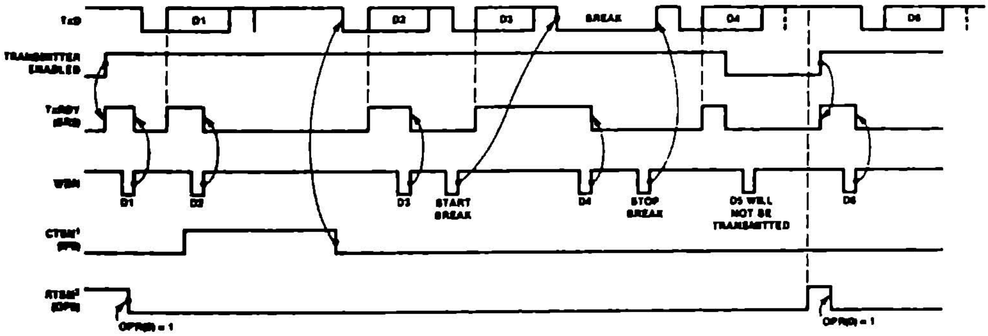

RxC
(1X INPUT)
tRXS
tRXH
RxD

FIGURE 7

TRANSMITTER TIMING

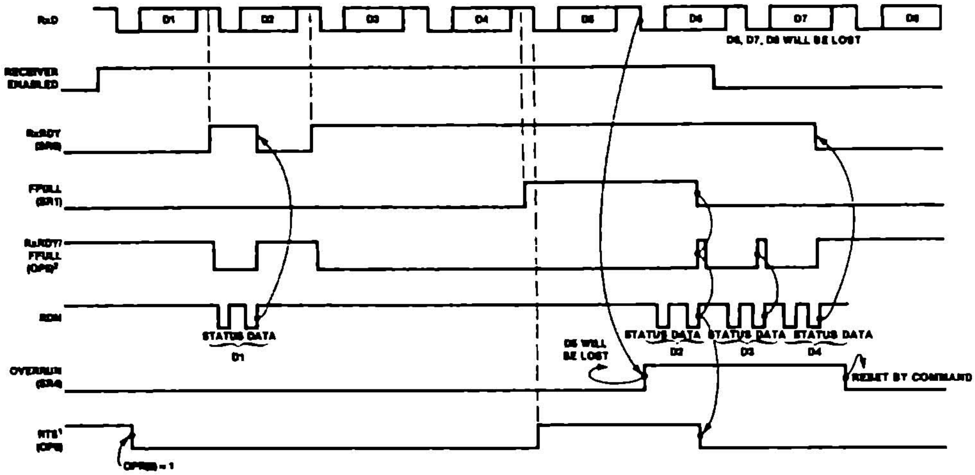

TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
WRN
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN¹
(IP0)
RTSN²
(OP0)
OPR(0) = 1
OPR(0) = 1

NOTES
1. TIMING SHOWN FOR MR2(4) = 1.
2. TIMING SHOWN FOR MR2(5) = 1.

FIGURE 8

Preview

RECEIVER TIMING


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFULL (SR1)
RxRDY/FFULL (OP5)²
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS¹ (OP0)
OPR(0) = 1

NOTES
1. TIMING SHOWN FOR MR1(7) = 1
2. SHOWN FOR OPCR(4) = 1 AND MR1(6) = 0

FIGURE 9

WAKE UP MODE

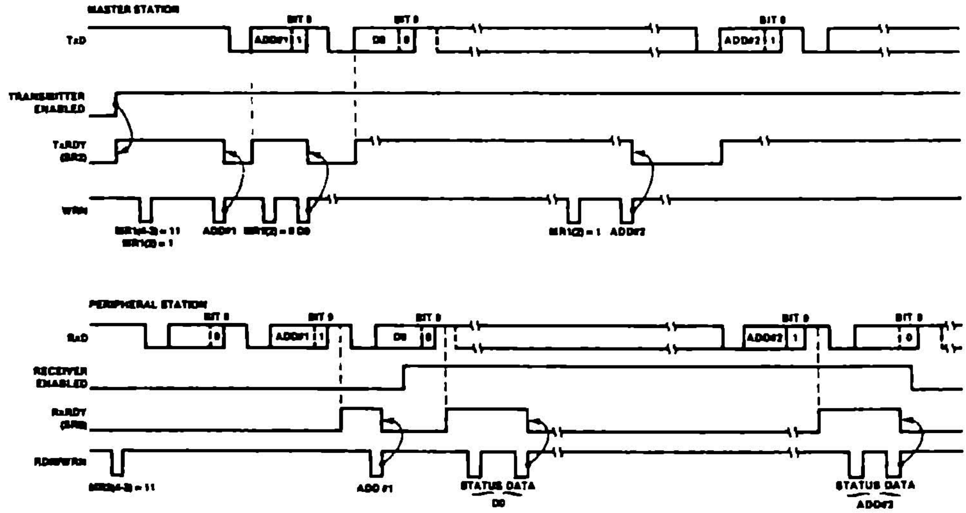

MASTER STATION
BIT 9
BIT 9
BIT 9
TxD
ADD#1
1
D0
0
ADD#2
1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1(4-3) = 11
MR1(2) = 1
ADD#1
MR1(2) = 0 D0
MR1(2) = 1
ADD#2
PERIPHERAL STATION
BIT 9
BIT 9
BIT 9
BIT 9
BIT 9
RxD
0
ADD#1
1
D0
0
ADD#2
1
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR1(4-3) = 11
ADD #1
STATUS DATA
D0
STATUS DATA
ADD#2

FIGURE 10

24-PIN ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI) SC2661

DESCRIPTION

The Signetics 24-pin Enhanced Programmable Communications Interface (EPCI) is a universal synchronous/asynchronous data communications controller chip. The EPCI serializes parallel data characters received from a microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microprocessor.

The 2661 contains a baud rate generator which can generate one of 16 transmit/receive clocks under program control. The device can also be programmed to use external transmit and/or receive clocks. The three versions of the 2661 (A, B, C) provide different baud rate sets. See table 1.

The 24-pin 2661 is functionally compatible with the 28-pin version of the device but is packaged in a slim (0.4" wide) 24-pin DIP. The reduced package area makes it particularly well suited for applications where multiple 2661s are required on a single board.

FEATURES

- **Synchronous operation**
  - **5 to 8-bit characters plus parity**
  - **Single or double SYN operation**
  - **Internal or external character synchronization**
  - **Transparent or non-transparent mode**
  - **Transparent mode DLE stuffing (Tx) and detection (Rx)**
  - **Automatic SYN or DLE-SYN insertion**
  - **SYN, DLE and DLE-SYN stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
- **Asynchronous operation**
  - **5 to 8-bit characters plus parity**
  - **1, 1½ or 2 stop bits transmitted**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode (echoplex)**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
    dc to 62.5K bps (16X clock)
    dc to 15.625K bps (64X clock)

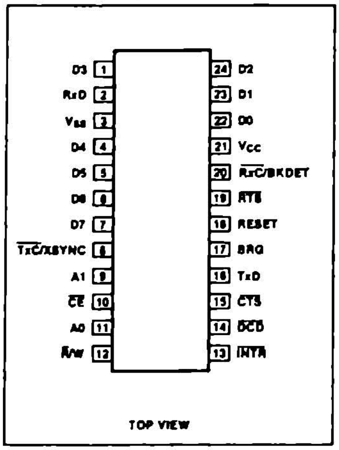
OTHER FEATURES

- **Internal or external baud rate clock**
- **3 baud rate sets**
- **16 internal rates for each set**
- **Double buffered transmitter and receiver**
- **Dynamic character length switching**
- **Full or half duplex operation**
- **TTL compatible inputs and outputs**
- **RxC and TxC pins are short circuit protected**
- **Single 5V power supply**
- **No system clock required**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front end processors**
- **Remote data concentrators**
- **Computer to computer links**
- **Serial peripherals**
- **BISYNC adaptors**

PIN CONFIGURATION



ORDERING CODE

PACKAGES	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $70°C$	
Ceramic DIP	SC2661ACSI24 SC2661BCSI24 SC2661CCSI24	See table 1 for baud rates
Plastic DIP	SC2661ACSN24 SC2661BCSN24 SC2661CCSN24	See table 1 for baud rates

PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
22-24, 1, 4-7	D0-D7	8-bit data bus	I/O
18	RESET	Reset	I
11, 9	A0-A1	Internal register select lines	I
12	$R/\overline{W}$	Read or write command	I
10	$\overline{CE}$	Chip enable input	I
19	$\overline{RTS}$	Request to send	O
15	$\overline{CTS}$	Clear to send	I
14	$\overline{DCD}$	Data carrier detected	I
13	$\overline{INTR}$	RxRDY, TxRDY, TxEMT, or DCD change interrupt output	O
8	$\overline{TxC}$/XSYNC	Transmitter clock/external sync	I/O
20	$\overline{RxC}$/BKDET	Receiver clock/break detect	I/O
16	TxD	Transmitter data	O
2	RxD	Receiver data	I
17	BRCLK	Baud rate generator clock	I
21	V_{CC}	+ 5V supply	I
3	GND	Ground	I

CHANGES FROM THE 28-PIN VERSION

The 24-pin 2661 is functionally similar to the 28-pin version of the device. The following differences apply to the 24-pin version:

1. The 24-pin version provides a single interrupt output ($\overline{INTR}$) instead of the three interrupt outputs (RxRDY, TxRDY, TxEMT/DSCHG) supplied on the 28-pin version. $\overline{INTR}$ will be asserted (low) when one or more of the status bits SR0, SR1 or SR2 is a logic one.
2. Two modem interface pins, the $\overline{DTR}$ output and the $\overline{DSR}$ input, have been eliminated. Because of this, status bit SR7 should be ignored and the setting of status bit SR2 due to a data set change (DSCHG) can be caused only by a change of the $\overline{DCD}$ input. Since the $\overline{DTR}$ output is eliminated, command register bit CR1 does not perform any function, although it remains writable and readable.

Other than the above, the functional operation, DC electrical characteristics, and AC electrical characteristics of the 24-pin version are identical to the 28-pin version. Consult the 28-pin version data sheet for additional information.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS

SC2661A (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6144
0001	75	1.2	-	4096
0010	110	1.7598	−0.01	2793
0011	134.5	2.152	-	2284
0100	150	2.4	-	2048
0101	200	3.2	-	1536
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	-	256
1010	1800	28.7438	−0.19	171
1011	2000	31.9168	−0.26	154
1100	2400	38.4	-	128
1101	4800	76.8	-	64
1110	9600	153.6	-	32
1111	19200	307.2	-	16

SC2661B (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	-	6144
0010	75	1.2	-	4096
0011	110	1.7598	−0.01	2793
0100	134.5	2.152	-	2284
0101	150	2.4	-	2048
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1200	19.2	-	256
1001	1800	28.7438	−0.19	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	-	128
1100	4800	76.8	-	64
1101	9600	153.6	-	32
1110	19200	307.2	-	16
1111	38400	614.4	-	8

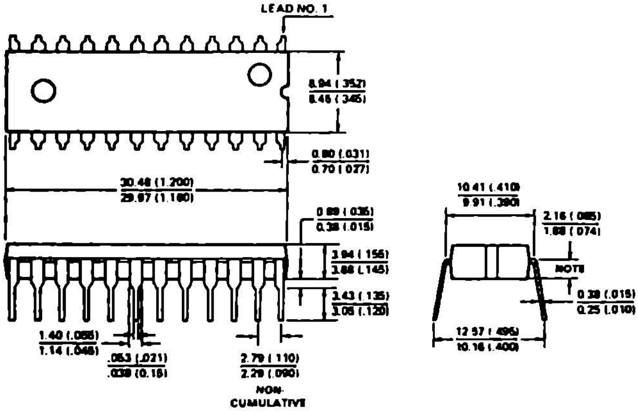
Table 1 BAUD RATE GENERATOR CHRACTERISTICS (Cont'd)
SC2661C (BRCLK = 5.0688MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6336
0001	75	1.2	-	4224
0010	110	1.76	-	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	-	2112
0101	300	4.8	-	1056
0110	600	9.6	-	528
0111	1200	19.2	-	264
1000	1800	28.8	-	176
1001	2000	32.081	0.253	158
1010	2400	38.4	-	132
1011	3600	57.6	-	88
1100	4800	76.8	-	66
1101	7200	115.2	-	44
1110	9600	153.6	-	33
1111	19200	316.8	3.125	16

NOTE
16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X and BRG can be used only for TxC

a

N PACKAGE — PLASTIC SLIM LINE
(24-PIN)


LEAD NO. 1
8.94 (.352)
8.46 (.345)
0.80 (.031)
0.70 (.027)
30.48 (1.200)
29.97 (1.180)
0.89 (.035)
0.38 (.015)
3.94 (.155)
3.68 (.145)
3.43 (.135)
3.05 (.120)
1.40 (.055)
1.14 (.045)
.053 (.021)
.038 (0.15)
2.79 (.110)
2.29 (.090)
NON-
CUMULATIVE
10.41 (.410)
9.91 (.390)
2.16 (.085)
1.88 (.074)
NOTE
0.38 (.015)
0.25 (.010)
12.57 (.495)
10.16 (.400)

NOTE
Lead spacing shall be measured within this zone
a. Shoulder and lead tip dimensions are to centerline of leads

JANUARY 1982

2661 OPERATING MODE SWITCHING PROCEDURES

App Note 402

INTRODUCTION

This application note describes procedures for switching the operating mode of the Signetics' 2661 Enhanced Programmable Communications Interface (EPCI) from echoplex or remote loopback mode to normal operation and vice-versa.

ECHOPLEX (AUTOMATIC ECHO) MODE TO NORMAL OPERATION

The echoplex operation is initiated by setting command register bits CR7:CR6 = 01, and CR2 (receiver enable bit) = 1. Echoplex operation is terminated by resetting CR2 to zero. To ensure the proper transmission of the last received character, no change of operating mode should be made until the end of that character. However, if mode switching is necessary in certain applications, the following procedure is recommended to ensure no garbling on the last transmitted character. Two potential problems may arise: the calculated parity instead of the received parity may be transmitted, and ata rate may be shortened or lengthened.

he procedure provides the necessary handshaking to avoid these potential problems by making use of the $\overline{\mathrm{TXEMT}}/\overline{\mathrm{DSCHG}}$ pin or of the status register bit 2, SR2, to indicate the end of the parity bit or the first stop bit, depending on whether one or two stop bits are selected (MR17:MR16 = 01 or 11). The procedure causes $\overline{\mathrm{TXEMT}/\mathrm{DSCHG}}$ to be driven to its active state only at the completion of the last character, as shown in figure 1.

The recommended sequence of operation is as follows:

1. Wait for RXRDY (either RXRDY interrupt or status read). This is necessary for the assembly of the last character to be completed and to ensure the transfer of this character to the transmitter.
2. Enable the transmitter by setting CR0 to one.
3. Disable the receiver by setting CR2 to zero.
4. Wait for TXEMT (either $\overline{\mathrm{TXEMT}}/\overline{\mathrm{DSCHG}}$ interrupt or status read). At this point, the parity bit or the first stop bit (if two stop bits are selected) has been sent out.
5. Change mode from echoplex to normal.
6. Load new character into the transmit holding register, THR. Further communication between the 2661 chip and the CPU will resume as normal - that is, $\overline{\mathrm{TXRDY}}$ is driven active to indicate that the THR is available for new data and $\overline{\mathrm{TXEMT}}$ is driven active upon underrun condition.

Note that the $\overline{\mathrm{TXEMT}}$ pin is not driven active in echoplex mode. It is optionally driven active when the above steps are followed, particularly the transmitter being enabled as indicated in step 2. Because the transmitter relies on CR0 = 1 and CR2 = 0 to drive $\overline{\mathrm{TXEMT}}$ active, it is necessary to set CR0 to zero in echoplex mode if it is desired not to drive $\overline{\mathrm{TXEMT}}$ active. CR0, transmitter enable, is ignored for data transmission in echoplex mode. It is, however, used to determine whether $\overline{\mathrm{TXEMT}}$ should be driven active.

If frequent mode switching is anticipated and it is desired to drive $\overline{\mathrm{TXEMT}}$ active, step 2 of the above procedure could be skipped, provided that the echoplex operation is initiated by enabling both the receiver and the transmitter - that is, CR2:CR0 = 11.

The $\overline{\mathrm{TXEMT}}$ timing shown above is only applicable when switching modes. Note that in normal operation, $\overline{\mathrm{TXEMT}}$ is driven active at the beginning of the last data bit or parity bit upon underrun condition.

REMOTE LOOP BACK MODE TO NORMAL OPERATION

The procedure is similar to the procedure for echoplex to normal, with the following exceptions:

1. No handshaking with RXRDY is required.
2. During step 3 of the previous procedure, CR2 goes to zero, and CR7:CR6 should be simultaneously changed from 11 to 01 (remote to echoplex). This is necessary because the logic implemented to drive $\overline{\mathrm{TXEMT}}$ active relies on echoplex information. However, this requirement does not need additional service from the controller because remote-to-echoplex switching is done at the same time as disabling the receiver.

NORMAL OPERATION TO ECHOPLEX OR REMOTE

To avoid garbling the last transmitted data, a mode switch from normal operation to echoplex or remote operation should be performed as follows:

1. Wait for TXEMT (either $\overline{\mathrm{TXEMT}}/\overline{\mathrm{DSCHG}}$ interrupt or status read) to be asserted.
2. Disable the transmitter by setting CR0 to zero.
3. Wait for TXEMT to be negated.
4. Change the mode from normal operation to echoplex or remote.

The timing is illustrated in figure 2.

$\overline{\mathrm{TXEMT}/\mathrm{DSCHG}}$ OPERATION FOR ECHOPLEX MODE SWITCHING

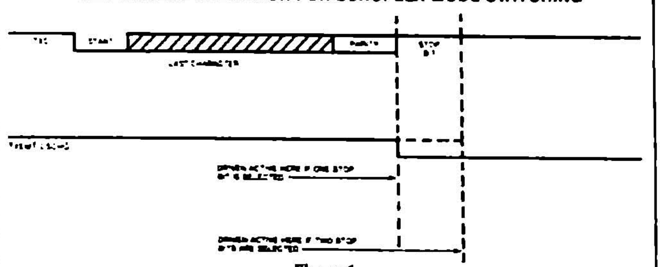

START
PARITY
STOP BIT
LAST CHARACTER
DRIVEN ACTIVE HERE IF ONE STOP BIT IS SELECTED
DRIVEN ACTIVE HERE IF TWO STOP BITS ARE SELECTED

Figure 1

SWITCHING FROM NORMAL TO ECHOPLEX OR REMOTE LOOPBACK

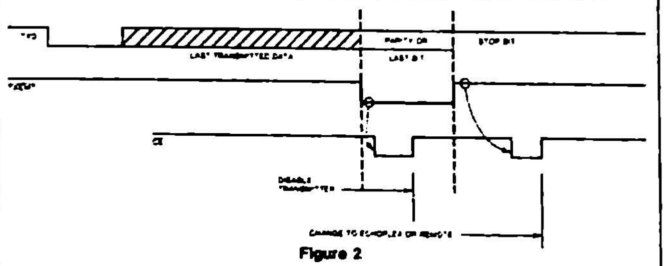


Figure 2

Section 2
Video Display

VALVO

JANUARY 1982

DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SC2670

Preliminary

DESCRIPTION

The Signetics Display Character and Graphics Generator (DCGG) is a mask-programmable 11,648-bit line select character generator. It contains 128 10X9 characters placed in a 10X16 matrix, and has the capability of shifting certain characters, such as j, y, g, p and q, that normally extend below the baseline. Character shifting, previously requiring additional external circuitry, is now accomplished internally by the DCGG; effectively, the 9 active lines are lowered within the matrix to compensate for the character's position.

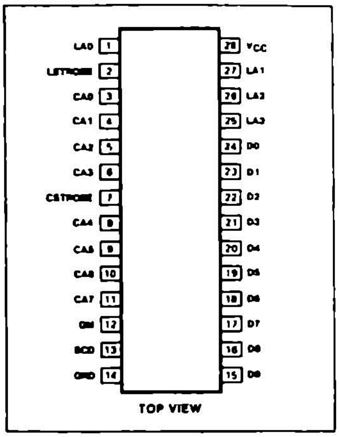
Seven bits of an 8-bit address code are used to select 1 of the 128 available characters. The eighth bit functions as a chip enable signal. Each character is defined by a pattern of logic ls and 0s stored in a 10X9 matrix. When a specific 4-bit binary line address code is applied, a word of 10 parallel bits appears at the output. The lines can be sequentially selected, providing a 9-word sequence of 10 parallel bits per word for each character selected by the address inputs. As the line address inputs are sequentially addressed, the device will automatically place the 10X9 character in 1 of 2 pre-programmed positions on the 16-line matrix with the positions defined by the 4-line address inputs. One or more of the 10 parallel outputs can be used as control signals to selectively enable functions such as half-dot shift, color selection, etc.

The 2670 DCGG includes latches to store the character address and line address data. A control input to inhibit character data output for certain groups of characters is also provided. The 2670 also includes a graphics capability, wherein the 8-bit character code is translated directly into 256 possible user programmable graphic patterns. Thus, the DCGG can generate data for 384 distinct patterns, of which 128 are defined by the mask programmable ROM. See figure 1 for a typical applications display.

FEATURES

- 128 10X9 matrix characters
- 256 graphic characters
- Optional thin graphics for forms
- Character and line address latches
- Internal descend logic
- 300nsec character select access maximum
- Control character output inhibit logic
- Static operation—no clocks required
- Single 5V power supply
- TTL compatible inputs and outputs

PIN CONFIGURATION

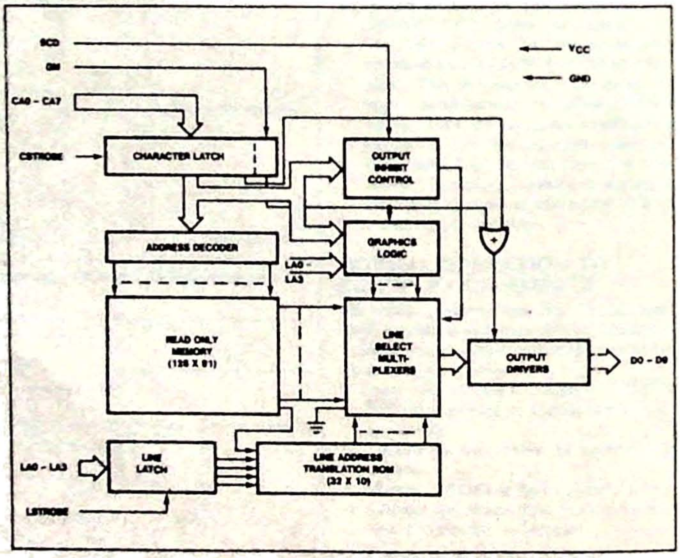

LA0 1
LSTROBE 2
CA0 3
CA1 4
CA2 5
CA3 6
CSTROBE 7
CA4 8
CA5 9
CA6 10
CA7 11
12
SCD 13
GND 14
28 VCC
27 LA1
26 LA2
25 LA3
24 D0
23 D1
22 D2
21 D3
20 D4
19 D5
18 D6
17 D7
16 D8
15 D9
TOP VIEW

ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ± 5%, T_A = 0°C to 70°C
Ceramic DIP	SC2670*CS128
Plastic DIP	SC2670*CSN28

NOTE
Substitute letter corresponding to standard font for * in part number for standard parts. See back of data sheet. Contact sales office for custom ROM patterns.

BLOCK DIAGRAM


SCD
CA0 - CA7
CSTROBE
CHARACTER LATCH
OUTPUT INHIBIT CONTROL
VCC
GND
ADDRESS DECODER
GRAPHICS LOGIC
LA0 - LA3
READ ONLY MEMORY (128 X 91)
LINE SELECT MULTI-PLEXERS
OUTPUT DRIVERS
D0 - D9
LA0 - LA3
LINE LATCH
LINE ADDRESS TRANSLATION ROM (32 X 10)
LSTROBE

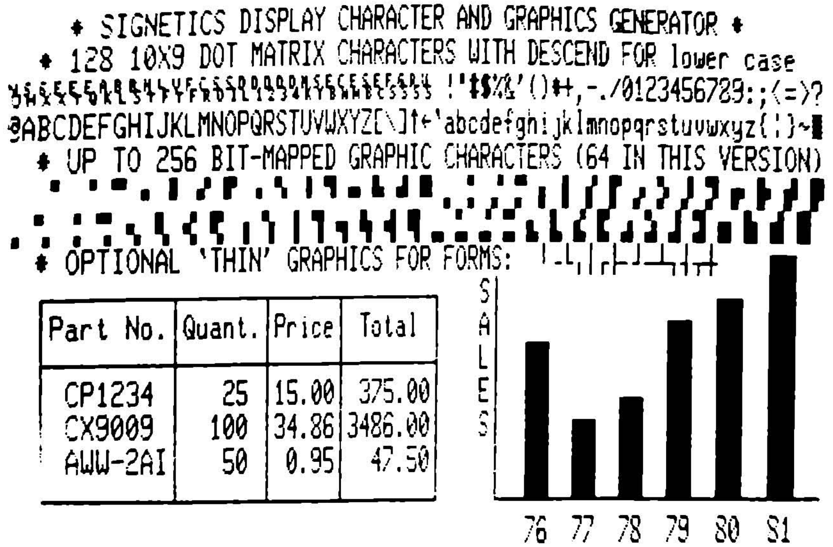
DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SC2670

Preliminary

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
CA0-CA7	3-6, 8-11	I	**Character Address:** Eight bit code specifies the character or graphic pattern for which matrix data is to be supplied. In character mode (GM=0), CA0 thru CA6 select one of the 128 ROM-defined characters and CA7 is a chip enable. The outputs are active when CA7=1 and are tri-stated when CA7=0. In graphics mode (GM=1), the outputs are active and CA0 thru CA7 select one of 256 possible graphic patterns to be output.
CSTROBE	7	I	**Character Strobe:** Used to store the character address (CA0 thru CA7) and graphics mode (GM) inputs into the character latch. Data is latched on the negative going edge of CSTROBE.
GM	12	I	**Graphics Mode:** GM=0 (low) selects character mode; GM=1 (high) selects graphics mode.
LA0-LA3	1, 25-27	I	**Line Address:** In character mode, selects one of the 16 lines of matrix data for the selected character to appear at the 10 outputs. LA0 is the LSB and LA3 is the MSB. The input codes which cause each of the nine lines of character data to be output are specified as part of the programming data for both non-shifted and shifted fonts. Cycling through the nine specified counts at the LA0 thru LA3 inputs cause successive lines of data to be output on D0 thru D9. The 7 non-specified codes for both non-shifted and shifted characters cause blanks (logic zeros) to be output. In graphics mode, the line address gates the latched graphics data directly to the outputs.
LSTROBE	2	I	**Line Strobe:** Used to store the line address data (LA0 thru LA3) in the line address latch. Data is latched on the negative going edge of LSTROBE.
SCD	13	I	**Selected Character Disable:** In character mode, a high level at this input causes all outputs (regardless of line address) to be blanks (zeros) for characters for which CA6 and CA5 are both 0. A low level input selects normal operation. Inoperative in the graphics mode.
D9-D0	15-24	O	**Data Outputs:** Provide the data for the specified character and line.
V_{CC}	28	I	+5V power supply.
GND	14	I	Ground.

TYPICAL APPLICATION



♦ SIGNETICS DISPLAY CHARACTER AND GRAPHICS GENERATOR ♦

♦ 128 10X9 DOT MATRIX CHARACTERS WITH DESCEND FOR lower case

♦ UP TO 256 BIT-MAPPED GRAPHIC CHARACTERS (64 IN THIS VERSION)

♦ OPTIONAL 'THIN' GRAPHICS FOR FORMS:

Part No.	Quant.	Price	Total
CP1234	25	15.00	375.00
CX9009	100	34.86	3486.00
AWW-2AI	50	0.95	47.50

Figure 1

DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SC2670

Preliminary

FUNCTIONAL DESCRIPTION

The DCGG consists of nine major sections. Line and character codes are strobed into the line and character latches. The character latch outputs are presented to the three sources of data: the ROM through an address decoder, the graphics logic, and the output inhibit control. The output inhibit control (together with the SCD input) suppresses the ROM data for selected character codes. The outputs from the line latch drive the line address translation ROM which maps the character ROM data onto 9 of 16 line positions. Finally, the line select multiplexers route the ROM or graphics data to the output drivers on D0 through D9.

Character Latch

The character latch is a 9-bit edge triggered latch used to store the character address (CA0 thru CA7) and graphics mode (GM) inputs. The data is stored on the falling edge of CSTROBE. Seven latched addresses (CA0 thru CA6) are inputs to the ROM character address decoder. In character mode (GM=0), CA7 operates as a chip enable. The output drivers are enabled when CA7=1 and are tri-stated when CA7=0. In graphics mode (GM=1), the output drivers are always enabled and the CA0 thru CA7 outputs of the latch are used to generate graphic symbols.

Character Address Decoder

This circuit decodes the 7-bit character address from the character latch to select one of the 128 character fonts stored in the ROM section of the DCGG.

Read Only Memory

The 11,648-bit ROM stores the fonts for the 128 matrix-defined characters. The data for each character consists of 91 bits. Ninety bits represent the 10X9 matrix and one bit specifies whether the character data is output at the normal (unshifted) lines or at the descended (shifted) lines. The 90 data bit outputs are supplied to the line select multiplexers. The descend control bit is an input to the line address translation ROM.

Graphics Logic

When the GM input is zero (low), the DCGG operates in the character mode. When it is one (high), it operates in the graphics mode. In graphics mode, output data is generated by the graphics logic instead of the ROM. The graphics logic maps the latched character address (CA0 thru CA7) to the outputs (D0 thru D9) as a function of line address (LA0 thru LA3). For any particular line address value, two of the CA bits are output: CA0, CA2, CA4 or CA6 is output on D0 thru D4 and CA1, CA3, CA5 or CA7 is output on D5 thru D9. The outputs are paired: When CA0 is output on D0 thru D4, CA1 is output on D5 thru D9 and likewise for CA2-CA3, CA4-CA5 and CA6-CA7.

A ROM within the graphics logic allows the specific line numbers for which each pair of bits is output to be specified by the customer. Figure 2 illustrates the general format for graphics symbols and an example where (CA7 thru CA0) = H'65'. The outputs from the graphics logic go to the line select multiplexers. The multiplexers route the graphic symbol data to the outputs when GM = 1.

Thin Graphics Option

As a customer specified option, 16 of the possible graphic codes (H'80' to H'8F') may be used to generate the special graphic characters illustrated in figure 3. For each of these characters, the vertical component appears on the D4 output. The horizontal component occurs on L_H which is specified by the customer. The vertical components specified by CA0 and CA2 are output for line addresses zero thru L_H and L_H thru fifteen, respectively.

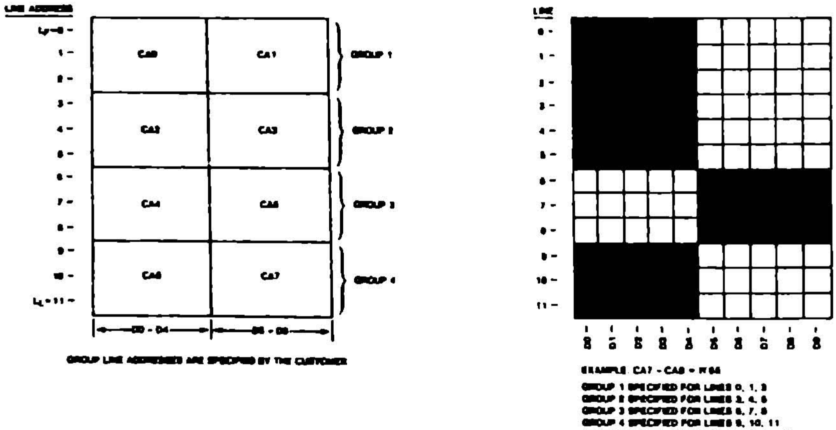
Line Select Multiplexers

The ten line select multiplexers select ROM data as specified by the line address translation ROM when GM=0, or graphics data when GM=1. The inputs to each multiplexer are the nine line outputs from the ROM, an output from the graphics logic and a logic zero (ground).

Output Drivers

Ten output drivers with 3-state capability serve as buffers between the line select multiplexers and external logic. The 3-state control input to these drivers is supplied from the CA7 latch when GM=0. When GM=1, the outputs are always active.

GRAPHICS SYMBOLS—GENERAL FORMAT



GROUP LINE ADDRESSES ARE SPECIFIED BY THE CUSTOMER

EXAMPLE: CA7 - CA0 = H'65

GROUP 1 SPECIFIED FOR LINES 0, 1, 2
GROUP 2 SPECIFIED FOR LINES 3, 4, 5
GROUP 3 SPECIFIED FOR LINES 6, 7, 8
GROUP 4 SPECIFIED FOR LINES 9, 10, 11
SPACE SPECIFIED FOR LINES 12, 13, 14, 15

Figure 2

Output Inhibit Control

The output inhibit control logic operates only if GM=0. It causes the output of the line select multiplexers to be logic zero if the SCD input is high and CA6 and CA5 of the latched character address are 00. If the SCD input is low, normal operation occurs. (This feature is useful in ASCII coded applications to selectively disable character generation for non-displayable characters such as line feed, carriage return, etc.)

Line Address Latch

The line address latch is a 4-bit latch used to store the line address (LA0-LA3). The data is stored on the negative edge of the LSTROBE input.

Line Address Translation ROM

This 32X10 ROM translates the 5-bit code consisting of the 4 outputs from the line address latch and the descend control bit from the ROM into a 1-of-10 code for the line select multiplexers. Programming information provided by the customer specifies the address which selects each line of ROM data for both shifted and non-shifted characters. Thus, there are nine line addresses which select ROM data for unshifted characters and nine addresses for shifted characters. These combinations are usually specified by the customer in either ascending or descending order. For the remaining 14 codes (7 each for unshifted and shifted characters), the translation ROM forces zeros at the outputs of the line select multiplexers.

This circuitry only operates if GM=0. When GM=1, the line select multiplexers are forced to select the outputs from the graphics logic.

Figure 4 shows an example of data outputs where the customer has specified line 14 as the first line for unshifted characters, line 11 as the first line for shifted characters and line address combinations in descending order.

CUSTOM PATTERN PROGRAMMING INSTRUCTIONS

A computer-aided technique utilizing punched computer cards is employed to specify a custom version of the 2670. This technique requires that the customer supply Signetics with a deck of standard 80-column computer cards describing the data to be stored in the ROM array, the programmable line address translation ROM, thin graphics option, and the graphics line font translation ROM.

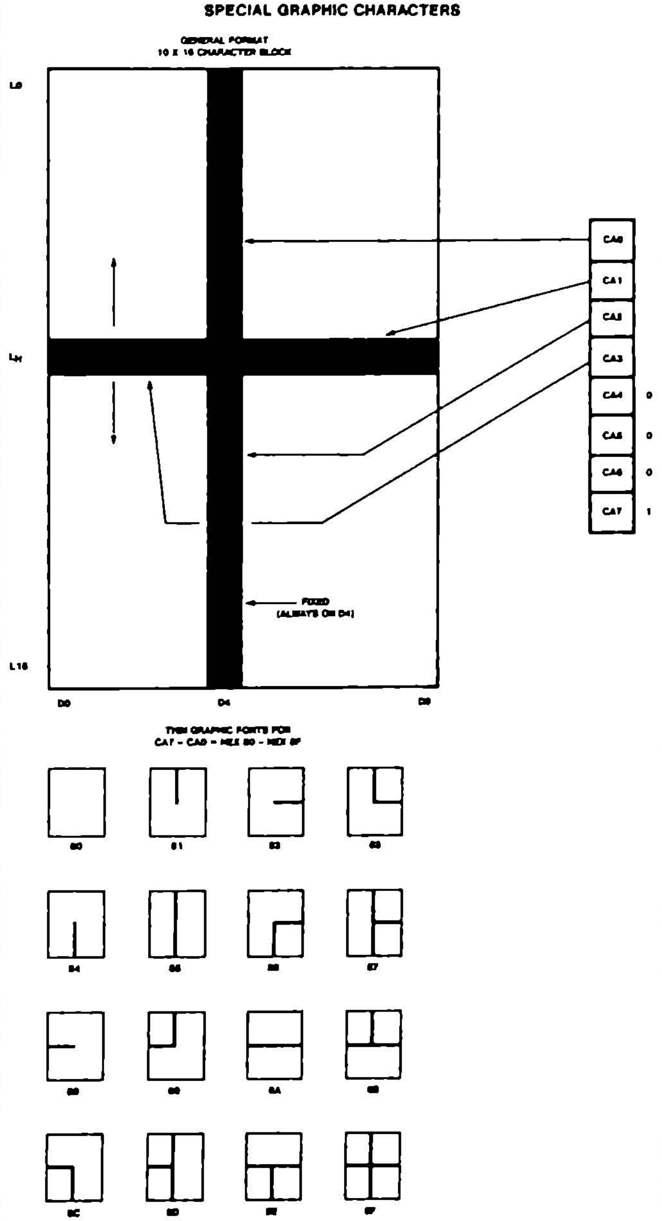


Figure 3

Preliminary

CUSTOMER SPECIFIED EXAMPLE

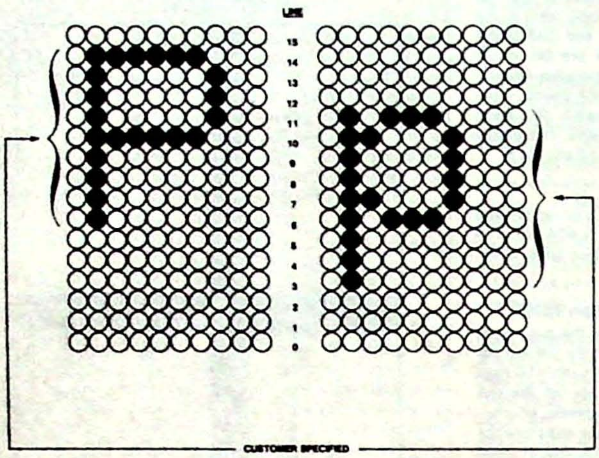


Figure 4

On receipt of a card deck, Signetics will translate the card deck to a truth table using the Signetics Computer Aided Design (CAD) facility. The truth table and font diagrams will then be sent to the customer for final approval. On receipt of final approval, Signetics will produce masks and proceed with manufacturing.

Programming information can also be input on TTY 7-level tape as card images. Each card image must be terminated with a carriage return-line feed. An EOT character must signify the end of the data set.

Customer identification cards are always labeled with a C in column 1. For customer identification, four cards are required. Any number of additional customer identification cards are permitted. The following data should be included:

CUSTOMER ID CARD #1

COLUMN	DATA
1	C
2	blank
3-9	2670/CP
10-14	blank
15-70	Company name/ company part number
71-80	blank

CUSTOMER ID CARD #2

COLUMN	DATA
1	C
2	blank
3-70	Customer contact person name/ phone number
71-80	blank

CUSTOMER ID CARD #3

COLUMN	DATA
1	C
2	blank
3-70	Customer address
71-80	blank

CUSTOMER ID CARD #4

COLUMN	DATA
1	C
2	blank
3-70	Customer city, state, zip code
71-80	blank

CUSTOMER ID CARD #5 THRU N

COLUMN	DATA
1	C
2	blank
3-70	Any information desired
71-80	blank

DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SC2670

Preliminary

The following masking information cards must be included:

Mask Information Card # 1: Shift and Nonshift Character Translation Data

COLUMN	DATA
1-9	NONSHIFT=
10	Line address in hex which outputs the first font word for nonshifted ROM fonts
11	,
12	Line address in hex which outputs the second font word for nonshifted ROM fonts
13	,
14	third
15	,
16	fourth
17	,
18	fifth
19	,
20	sixth
21	,
22	seventh
23	,
24	eighth
25	,
26	ninth
27-29	blank
30-35	SHIFT=
36	Line address in hex which outputs the first font word for shifted ROM fonts
37	,
38	second
39	,
40	third
41	,
42	fourth
43	,
44	fifth
45	,
46	sixth
47	,
48	seventh
49	,
50	eighth
51	,
52	ninth
53-59	blank
60[1]	0 or 1
61-64	blank
65[2]	0 or 1
66-80	blank

NOTES

1. Column 60 specifies the font truth table horizontal format. 0 specifies left to right printing of D0 thru D9. 1 specifies D9 thru D0.
2. Column 65 specifies the font truth table vertical printout format. 0 specifies top to bottom printing of line address hex 0 thru F. 1 specifies hex F thru 0.

MASK INFORMATION CARD #2: Graphics Translation Data

COLUMN	DATA
1-14	THIN GRAPHICS=
15-17	YES or NOb̸, where b̸ = blank. Specifies whether graphics address hex 80 thru hex 8F will select the special thin graphics font.
18-19	blank
20-23	HOR=
24	The line address in hex for the horizontal segments of line graphics fonts. Leave blank if columns 15 thru 17 are NO
25-29	blank
30-45	Graphics group number 1 or 2 or 3 or 4 or blank. Columns 30 thru 45 correspond to line address hex 0 thru hex F respectively. The group number specified in each column will cause the graphics data generated by that group to be output at the corresponding line address. A blank specifies no data for that address.
46-80	blank

MASK INFORMATION CARD #3 THRU #130: ROM Font Data

COLUMN	DATA
1-2	Character address in hex (CA6 thru CA0)*
3	blank
4	S for shifted; N for nonshifted.
5	blank
6-8	Data for first ROM font word in hex (D9 thru D0).
9	blank
10-12	second
13	blank
14-16	third
17	blank
18-20	fourth
21	blank
22-24	fifth
25	blank
26-28	sixth
29	blank
30-32	seventh
33	blank
34-36	eighth
37	blank
38-40	ninth
41-80	blank

NOTE

*A separate card is required for each character address hex 00 thru hex 7F

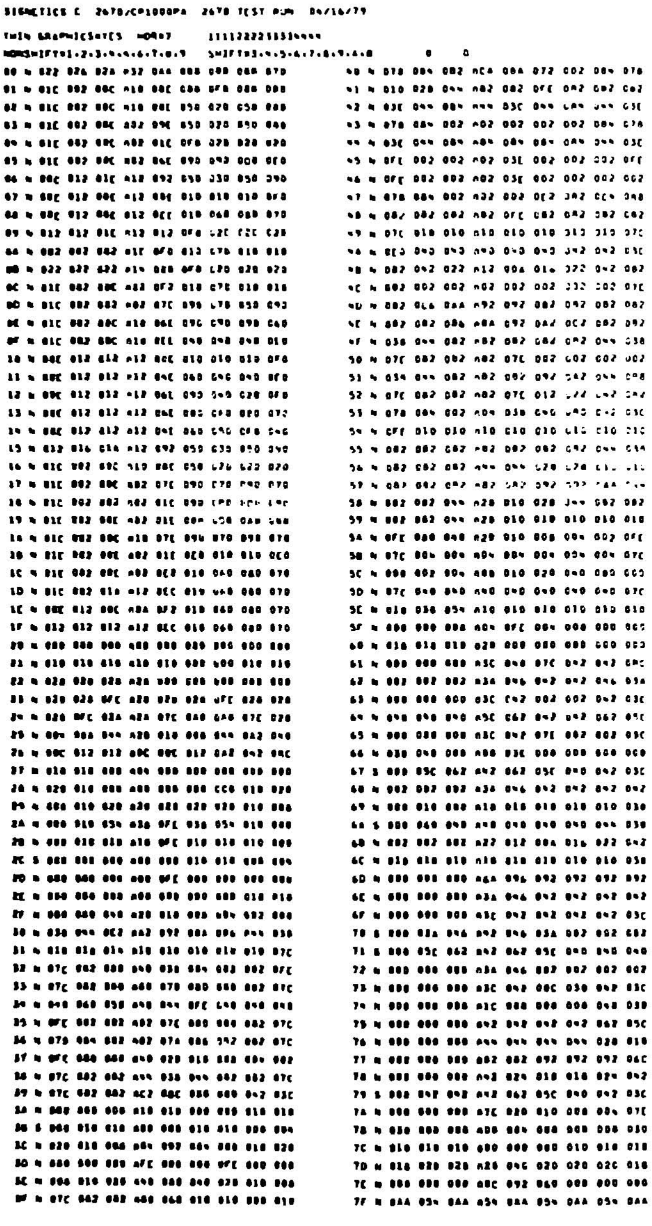
Preliminary

Printouts

Signetics will translate the card deck to the following printouts to be submitted to the customer for approval:

- A repeat of all customer information.
- A separate font drawing for each of the 128 ROM characters and 256 graphics fonts. The font drawings are positioned on a 10 X 16 matrix as specified by the customer's translation data.

SAMPLE CARD DECK INPUT



DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG)

SC2670

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltage	6.0	V
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.3 to +6.0	V

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless it is suggested that conventional precautions be taken to avoid applying any voltages larger than the maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5.0V \pm 5\%$ [1,2,3]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage		0		0.8	V
V_{IH}	Input high voltage		2.0		V_{CC}	V
V_{OL}	Output low voltage	$I_O = 1.6mA$	0		0.45	V
V_{OH}	Output high voltage	$I_O = -100\mu A$	2.4		V_{CC}	V
I_{IL}	Input leakage current	$V_{IN} = 0$ to $4.25V$			10	μA
I_{OL}	Output leakage current	$V_O = 0.4$ to $4V$			±10	μA
I_{CC}	Supply current	$V_{CC} = 5.25V$		50	100	mA
C_{IN}	Input capacitance	All other pins grounded			10	pF
C_{OUT}	Output capacitance				15	pF

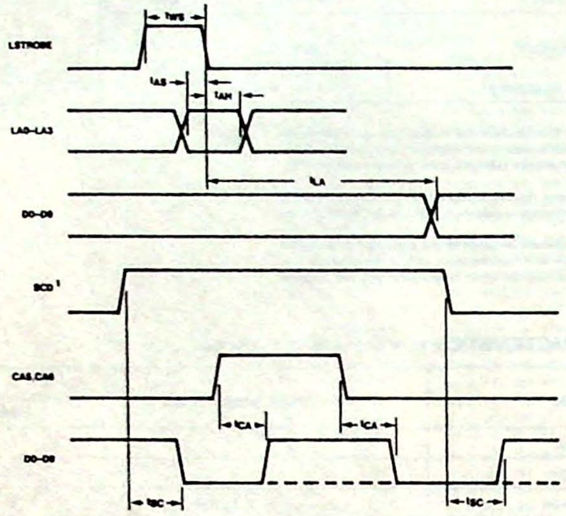
AC CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$ [1,2,3]

PARAMETER		TENTATIVE LIMITS		UNIT
		Min	Max	
t_{WS}	Strobe pulse width	100		ns
t_{AS}	Address setup	50		ns
t_{AH}	Address hold	25		ns
t_{CA}	Character select access		300	ns
t_{LA}	Line select access		500	ns
t_{SEL}	Chip select delay[4]		250	ns
t_{DES}	Chip deselect delay[4]		200	ns
t_{SC}	Special character blank/unblank time		1	μs

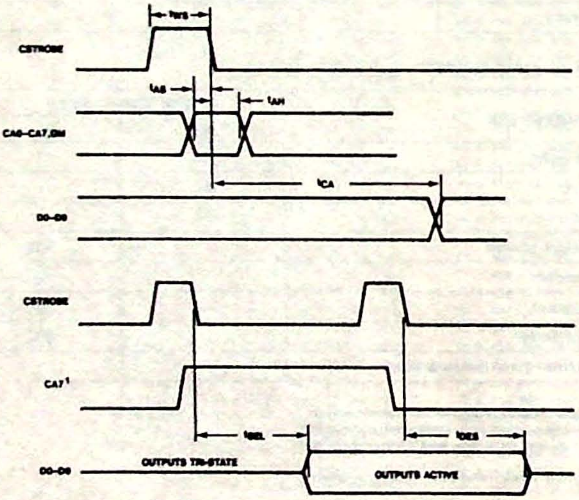
NOTES

1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at 50% level for inputs and at the 0.8V or 2.0V level for outputs. Input levels are 0.45V and 2.4V.
3. Typical values are at +25°C, typical supply voltages and typical processing parameters.
4. Test conditions: $C_L = 100pF$ and 1 TTL load.

TIMING DIAGRAMS

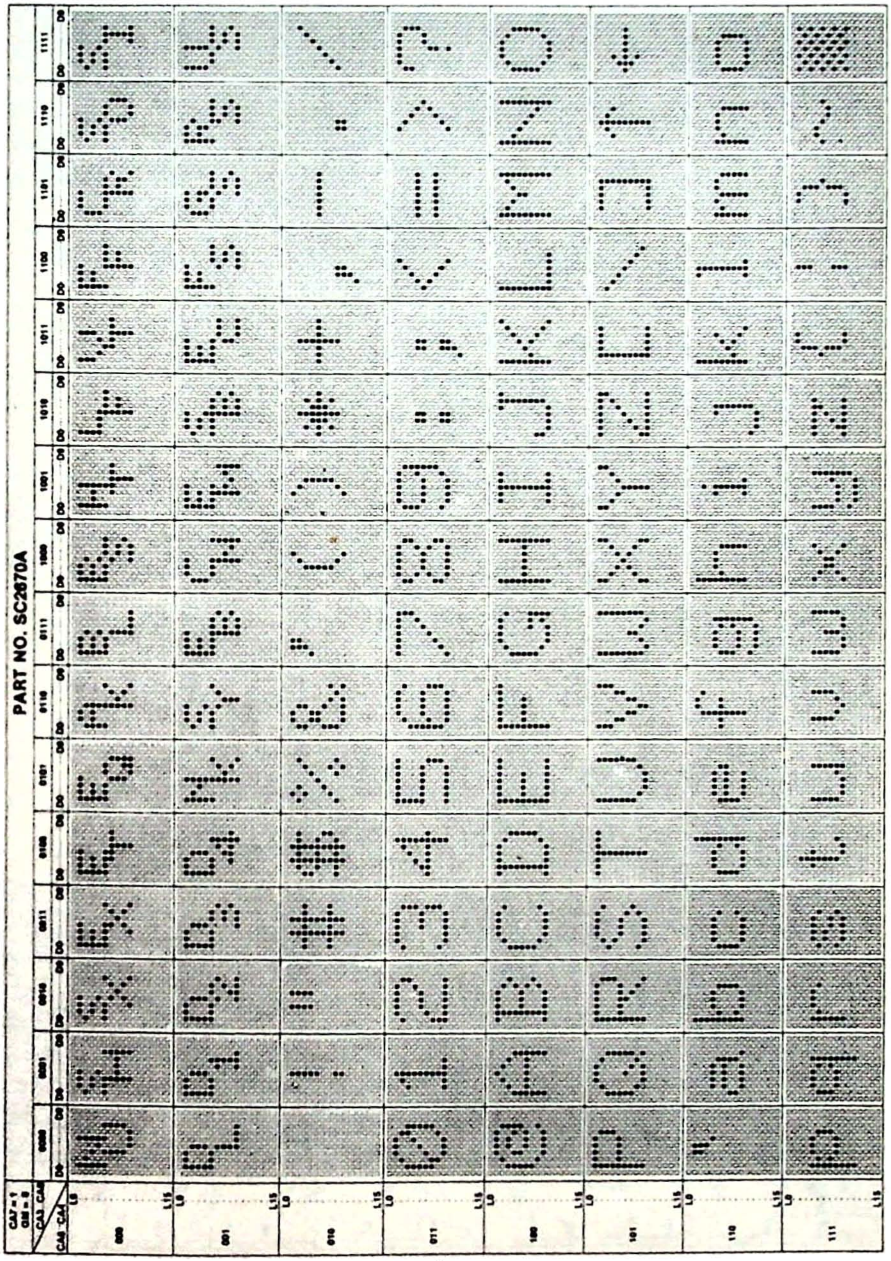

tWS
LSTROBE
tAS
tAH
LA0–LA3
tLA
D0–D8
SCD1
CA5,CA6
tCA
tCA
D0–D8
tSC
tSC

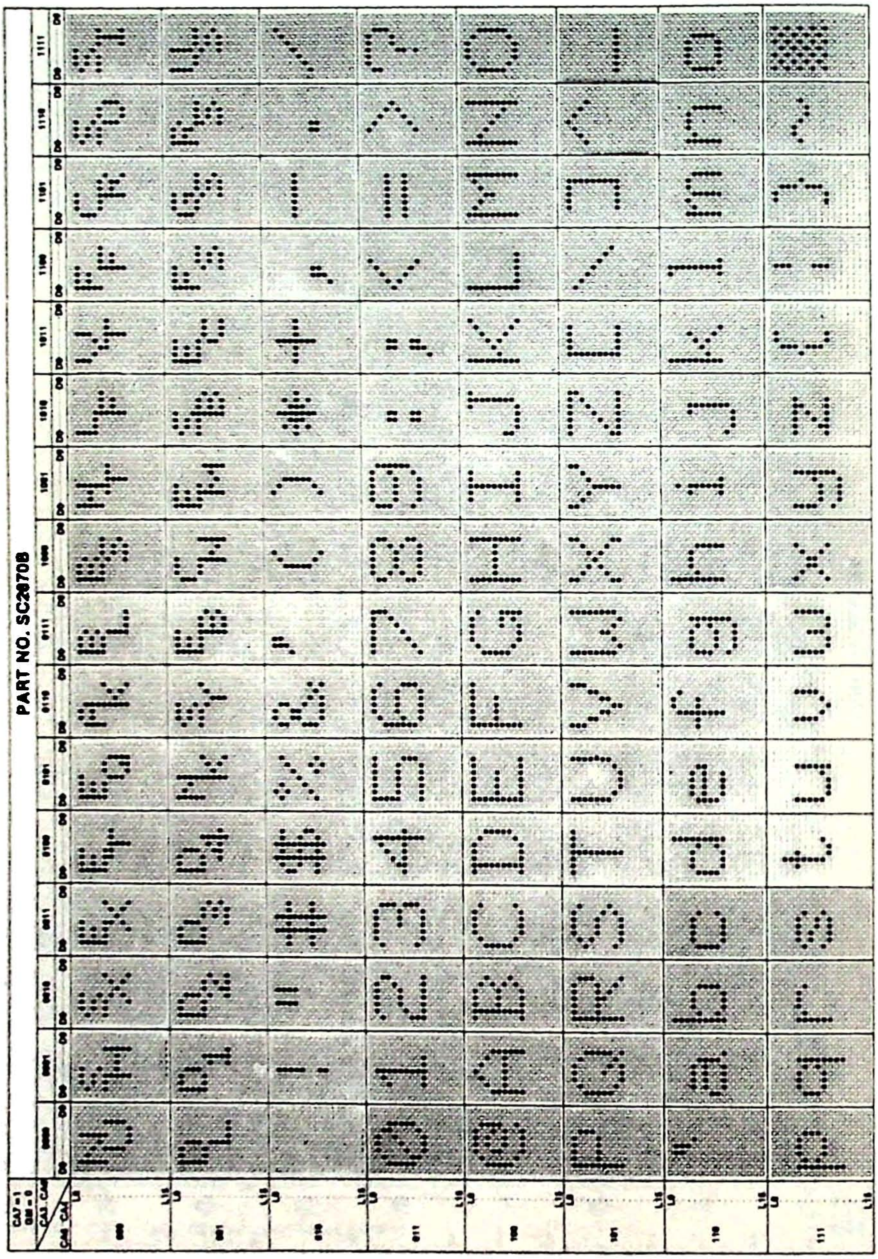
NOTE
1. WHEN GM=1, SCD INPUT IS INACTIVE


tWS
CSTROBE
tAS
tAH
CA0–CA7,GM
tCA
D0–D8
CSTROBE
CA71
tSEL
tDES
OUTPUTS TRI-STATE
D0–D8
OUTPUTS ACTIVE

NOTE
1. CA7 OPERATES AS OUTPUT ENABLE ONLY IN CHARACTER MODE (GM=0)

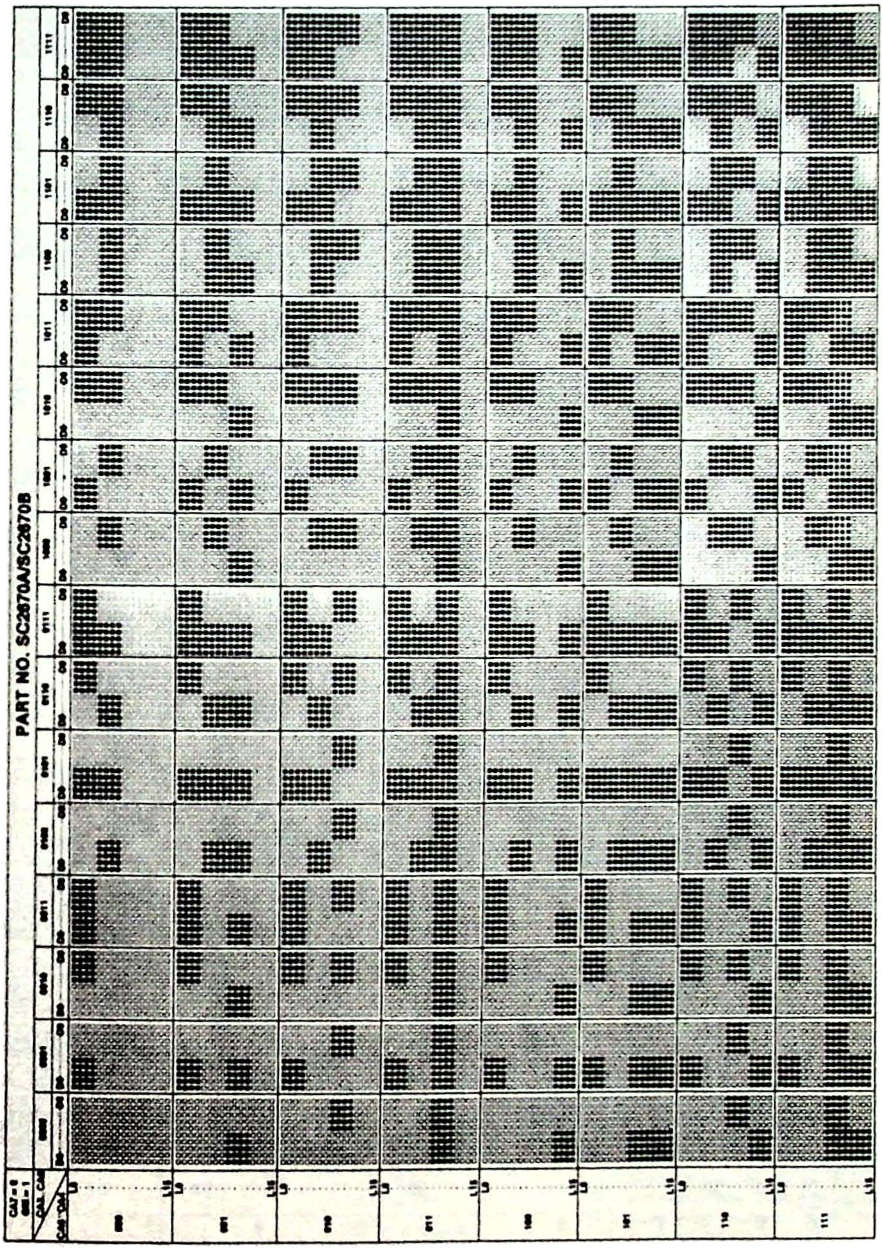
Preliminary


PART NO. SC2670A
CA7 = 1
GM = 0
CA3 - CA0
CA6 - CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
000
001
010
011
100
101
110
111
L0
L15
D8
D0


PART NO. SC2670B

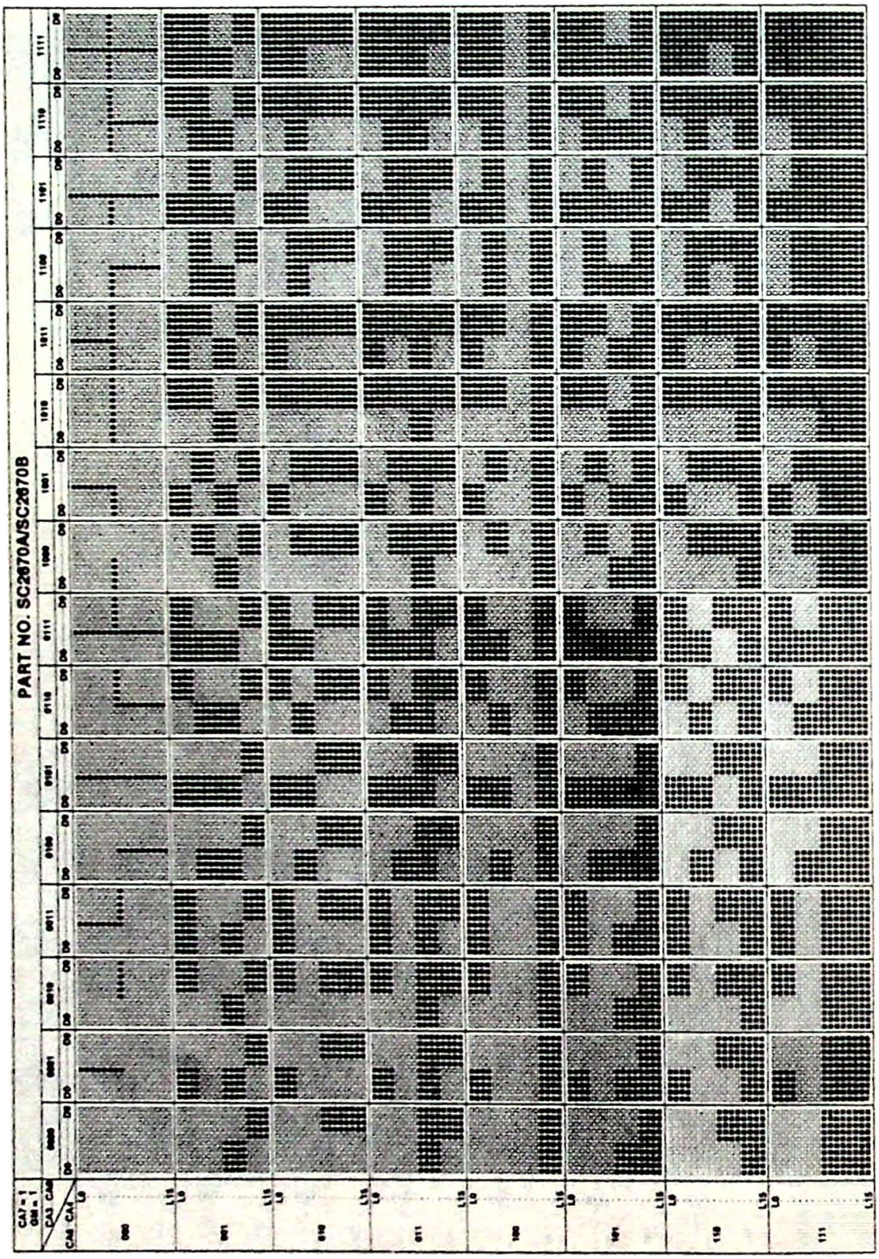
DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SC2670

Preliminary


PART NO. SC2670A/SC2670B
1111
1110
1101
1100
1011
1010
1001
1000
0111
0110
0101
0100
0011
0010
0001
0000
000
001
010
011
100
101
110
111

DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SC2670

Preliminary


PART NO. SC2670A/SC2670B
CA7 = 1
GM = 1
CA3 CA0
CA6 CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D9
L0
L15
000
001
010
011
100
101
110
111

JANUARY 1982

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

DESCRIPTION

The Signetics 2671 Programmable Keyboard and Communications Controller (PKCC) is an MOS LSI device which provides a versatile keyboard encoder and an independent full duplex asynchronous communications controller. It is intended for use in microprocessor based systems and provides an eight bit data bus interface.

The keyboard encoder handles the scanning, debounce, and encoding of a keyboard matrix with a maximum of 128 keys. It provides four levels of key encoding corresponding to the separate SHIFT and CONTROL input combinations. Four keyboard rollover modes can be programmed including provisions for up to 16 latched keys. Control outputs are provided for interfacing with contact or capacitive keyboards. An eight bit keyboard status register provides status information to the CPU.

The receiver section of the communications controller accepts serial data from the RxD pin and converts it to parallel data characters. Simultaneously, the transmitter section accepts parallel data from the data bus and outputs serialized data onto the TxD pin. Received data is checked for parity and framing errors, and break conditions are flagged. Character lengths can be programmed as 5, 6, 7, or 8 bits not including parity, start or stop bits. An internal baud rate generator (BRG) with 16 divider ratios can be used to derive the receive and/or transmit clocks. The BRG can accept an external clock or operate directly from a crystal. An eight bit communications status register provides status information to the CPU.

The PKCC has an interrupt mask register to selectively enable certain keyboard and communications status bits to generate interrupts. Priority encoded interrupt vectoring is available. Upon receipt of an interrupt acknowledge, an interrupt vector will be output on D0-D7 reflecting the source of the interrupt. The interrupt source can also be read from an interrupt status register.

ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ±5%, T_A = 0°C to 70°C
Ceramic DIP	SC2671ACSI40
Plastic DIP	SC2671ACSN40

FEATURES

- Keyboard interface
  - Contact or capacitive keyboard
  - Up to 128 keys on an 8 X 16 matrix
  - Encoded or unencoded operation
  - Four code levels per key
  - Latched key option—separate depress and release codes
  - Programmable scan rate and debounce time
  - Programmable rollover modes
  - Programmable auto-repeat for selected keys
  - Tone output—two frequencies
- Asynchronous communication interface
  - Internal baud rate generator—16 rates
  - Full duplex operation
  - Detection of start and end of break
  - Programmable break generation
  - Programmable character parameters
  - Auto-echo and maintenance loopback modes
- Polled or interrupt operation
- Interrupt priority controller and vector generator
- Operates directly from crystal or external clocks
- TTL compatible
- Single +5 volt power supply
- 40 pin dual in-line package

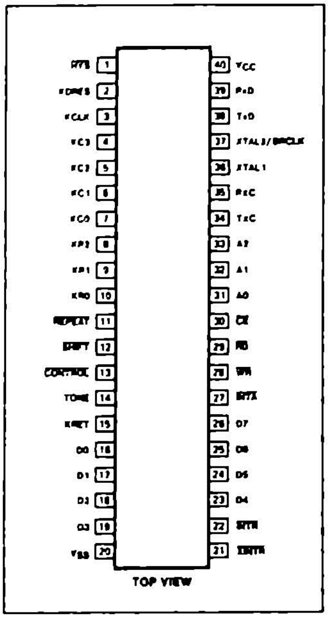
APPLICATIONS

- CRT terminals
- Hard copy terminals
- Word processing systems
- Data entry terminals
- Small business computers

FUNCTIONAL DESCRIPTION

The PKCC consists of six major sections (see block diagram). These are the transmitter, receiver, timing, operation control, keyboard encoder, and a priority encoded interrupt control unit. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a bidirectional data bus buffer.

PIN CONFIGURATION



Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers KMR and CMR, the command decoder, and status registers KSR and CSR. Details of operating modes and status information are presented in the Operation section of this data sheet. The register addressing is specified in table 1.

Timing

The PKCC contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 baud rates, any of which can be selected for full duplex operation. The external clock to the baud rate generator can be applied directly to the XTAL2 input (see figure 21) or can be generated internally by connecting a crystal across the XTAL1, XTAL2 input pins. The clock input is also utilized by the keyboard encoder section. Thus, a clock must be provided even if external transmitter and receiver clocks are used.

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0-D7	16-19, 23-26	I/O	**Data Bus:** 8-bit three-state bidirectional data bus. All data, command and status transfers are made using this bus. D0 is the least significant bit; D7 is the most significant bit.
A0-A2	31-33	I	**Address Lines:** Used to select internal PKCC registers or commands.
$\overline{RD}$	29	I	**Read Strobe:** When low, gates the selected PKCC register onto the data bus if $\overline{CE}$ is also low.
$\overline{WR}$	28	I	**Write Strobe:** When low, gates the contents of the data bus into the selected PKCC register if $\overline{CE}$ is also low.
$\overline{CE}$	30	I	**Chip Enable:** When high, places the D0-D7 output drivers in a three-state condition. If $\overline{CE}$ is low, data transfers are enabled in conjuction with the $\overline{RD}$ and $\overline{WR}$ inputs.
$\overline{INTR}$	22	O	**Interrupt Request:** Several conditions may be programmed to request an interrupt to the CPU. It is an active low open-drain output. This pin will be inactive after power on reset or a master reset command.
$\overline{INTA}$	27	I	**Interrupt Acknowledge:** Used to indicate that an interrupt request has been accepted by the CPU. When $\overline{INTA}$ goes low, the PKCC outputs an 8-bit address vector on D0-D7 corresponding to the highest priority interrupt currently active.
$\overline{XINTR}$	21	I	**External Interrupt:** An active low external interrupt input to the PKCC interrupt priority resolver.
TxC	34	I/O	**Transmitter Clock:** The function of this pin depends on bit 7 of the baud rate control register (BRR7). If external transmitter clock is selected (BRR7 = 0), it is an input for the transmitter clock. If internal transmitter clock is selected (BRR7 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR5. The data is transmitted on the falling edge of TxC. It is an input after power on and after master reset or communications reset commands.
RxC	35	I/O	**Receiver Clock:** The function of this pin depends on BRR6. If external receiver clock is selected (BRR6 = 0), it is an input for the receiver clock. If internal receiver clock is selected (BRR6 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR4. The received data is sampled on the rising edge of RxC. It is an input after power on and after master reset or communications reset commands.
TxD	38	O	**Transmitter Data:** This output is the transmitted serial data; the least significant bit is transmitted first. This pin is high after power on reset or a reset command that affects the transmitter.
RxD	39	I	**Receiver Data:** This input is the serial data input to the receiver. The least significant bit is received first.
XTAL 1 XTAL2/BRCLK	36,37	I	**Connections for Crystal:** Provides an on-chip clock generator for the internal baud rate generator and the keyboard interface logic. If an external clock is provided, use XTAL2 as the clock input. See figures 20 and 21. All timing parameters such as keyboard scan time, tone frequency, and baud rate assume a clock input at the specified BRG input frequency. If this frequency is different, the timing parameters will vary proportionately.
KR0-KR2	10-8	O	**Keyboard Row Scan:** Decoded externally; selects one of eight rows.
KC0-KC3	7-4	O	**Keyboard Column Scan:** Decoded externally; selects one of 16 columns.
KRET	15	I	**Key Return:** An active high level indicates that the key being scanned is closed.
$\overline{SHIFT}$	12	I	**SHIFT Key:** Active low input from the SHIFT key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs select one of four possible codes from the internal key encoding ROM.
$\overline{CONTROL}$	13	I	**CONTROL Key:** Active low input from the CONTROL key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs select one of four possible codes from the internal key encoding ROM.

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

PIN DESIGNATION (Cont.)

$\overline{\text{REPEAT}}$	11	I	**REPEAT Key:** Active low input from the $\overline{\text{REPEAT}}$ key. Causes the key depression currently active to be repeated at a rate of approximately 15 times per second.
KCLK	3	O	**Keyboard Clock:** High frequency (approximately 400 kHz) output used to scan capacitive keyboards.
KDRES	2	O	**Key Detect Reset:** Resets the analog detector before scanning a key. Used for capacitive keyboards.
$\overline{\text{HYS}}$	1	O	**Hysteresis Output:** Sent to the analog detector for capacitive keyboard applications. A low indicates the key currently being scanned has been recognized on previous scan cycles.
TONE	14	O	**Square Wave Output:** Used for tone generation.
V_{CC}	40	I	+5V power supply.
V_{SS}	20	I	Ground.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for break conditions, framing and parity errors, and loads an "assembled" character in the receive holding register for access by the CPU.

Transmitter

The transmitter accepts parallel data loaded by the CPU into the transmit holding register and converts it to a serial bit stream framed by the start bit, calculated parity bit (if specified), and stop bit(s). The composite serial stream of data is transmitted on the TxD output pin.

Keyboard Encoder

The keyboard encoder provides encoded scanning signals for a matrix keyboard. Key depressions are detected on the KRET input. The debounced and verified key codes (or matrix addresses) are loaded into the key holding register for access by the CPU. Figures 1 and 2 illustrate the PKCC interface to contact and capacitive keyboards, respectively.

Interrupt Control

The interrupt controller unit contains a software programmable interrupt mask register which selectively enables status conditions from the keyboard encoder and communication controller to generate interrupts. The interrupts are priority encoded and individually generate an eight bit vector which is output on the data bus in response to a CPU interrupt acknowledge on the $\overline{\text{INTA}}$ input pin.

OPERATION

Keyboard Encoder

The keyboard is continuously scanned by KC0-KC3 and KR0-KR2 which are decoded externally to handle 128 possible keys (see figures 1 and 2). KC0-KC3 select one of 16 columns and KR0-KR2 multiplex the eight row return lines into the KRET pin. Debouncing is accomplished by remembering a 1 state at the KRET pin when a key is being addressed and verifying it one scan later. Once the key is verified, a key code is loaded into the keyboard data register (KDR). If the keyboard holding register (KHR) is empty, the contents of the KDR will be transferred to the KHR immediately; if the KHR is full (i.e., the CPU has not read the previous key code), the transfer will be held off until the KHR is read. The data transfer to the KHR causes keyboard data ready (KRDY) to be set in the keyboard status register.

For capacitive keyboards, the high frequency output KCLK can be used to gate the column scan to the keyboard (see figure 2). The key detector reset (KDRES) output resets the analog detector prior to scanning each key location. The output from the analog multiplexer is sensed and then latched in the analog detector. The $\overline{\text{HYS}}$ output controls the sense level. A 0 will lower the sense level causing hysteresis, and a 1 will raise the sense level with no hysteresis.

The $\overline{\text{REPEAT}}$ input enables the keyboard logic to recognize any key repeatedly, 15 times per second. Additionally, certain keys can be programmed to repeat automatically if depressed for more than one-half second.

A square wave is output on the TONE pin when the CPU issues a ring tone command to the PKCC.

$\overline{\text{CE}}$	A2	A1	A0	$\overline{\text{RD}}/\overline{\text{WR}}$	FUNCTION
1	X	X	X	X	Three-state data bus
0	0	0	0	$\overline{\text{WR}}$	Reset command (see table 6)
0	0	0	0	$\overline{\text{RD}}$	Read interrupt status register (ISR)
0	0	0	1	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write communications mode register (CMR)
0	0	1	0	$\overline{\text{WR}}$	Write transmit holding register (TxHR)
0	0	1	0	$\overline{\text{RD}}$	Read receiver holding register (RxHR)
0	0	1	1	$\overline{\text{WR}}$	Write baud rate mode register (BRR)
0	0	1	1	$\overline{\text{RD}}$	Read communications status register (CSR)
0	1	0	0	$\overline{\text{RD}}$,$\overline{\text{WR}}$	Read/write interrupt mask register (IMR)
0	1	0	1	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write keyboard mode register (KMR)
0	1	1	0	$\overline{\text{RD}}$	Read keyboard holding register (KHR)
0	1	1	1	$\overline{\text{RD}}$	Read keyboard status register (KSR)
0	1	1	1	$\overline{\text{WR}}$	Miscellaneous commands (see description)

NOTE
X = don't care.

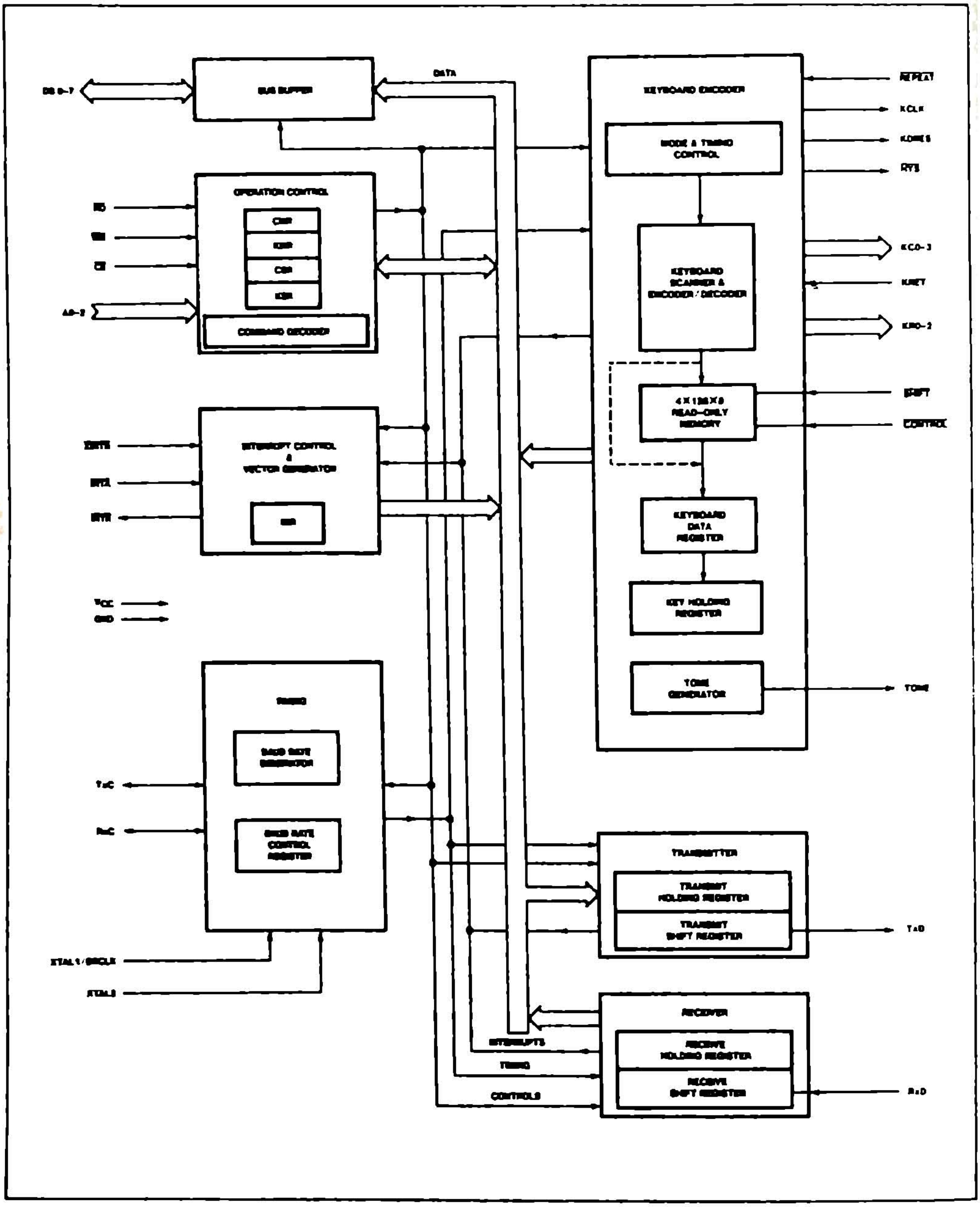
Table 1. 2671 REGISTER ADDRESSING

JANUARY 1982

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

BLOCK DIAGRAM


DB0-7
BUS BUFFER
DATA
OPERATION CONTROL
CMR
KMR
CSR
KSR
COMMAND DECODER
RD
WR
CE
A0-2
INTERRUPT CONTROL & VECTOR GENERATOR
INTA
INTR
VCC
GND
TIMING
BAUD RATE GENERATOR
BAUD RATE CONTROL REGISTER
TxC
RxC
XTAL1/BRCLK
XTAL2
KEYBOARD ENCODER
MODE & TIMING CONTROL
KEYBOARD SCANNER & ENCODER / DECODER
4 X 128 X 8 READ-ONLY MEMORY
KEYBOARD DATA REGISTER
KEY HOLDING REGISTER
TONE GENERATOR
REPEAT
KCLK
RTS
KC0-3
KRET
KR0-2
SHIFT
CONTROL
TONE
TRANSMITTER
TRANSMIT HOLDING REGISTER
TRANSMIT SHIFT REGISTER
TxD
RECEIVER
RECEIVE HOLDING REGISTER
RECEIVE SHIFT REGISTER
RxD
INTERRUPTS
TIMING
CONTROLS

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

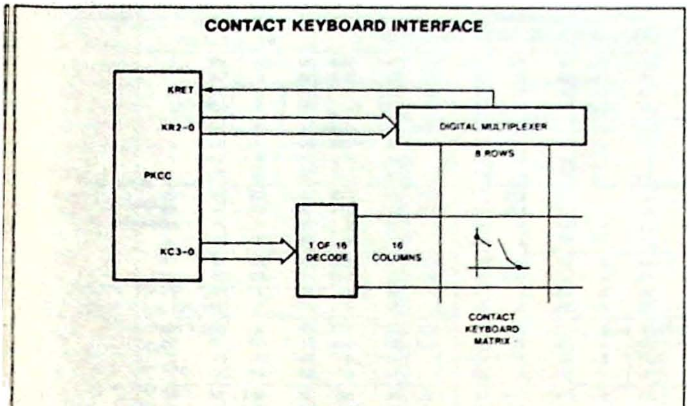

CONTACT KEYBOARD INTERFACE
KRET
KR2-0
PKCC
KC3-0
DIGITAL MULTIPLEXER
8 ROWS
1 OF 16 DECODE
16 COLUMNS
CONTACT KEYBOARD MATRIX

Figure 1

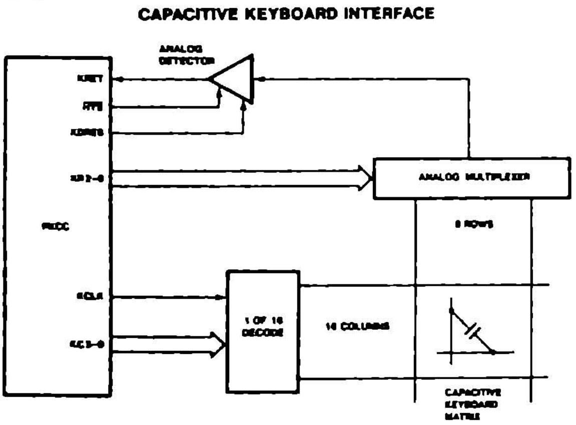

CAPACITIVE KEYBOARD INTERFACE
ANALOG DETECTOR
KRET
KR2-0
PKCC
KCLK
KC3-0
ANALOG MULTIPLEXER
8 ROWS
1 OF 16 DECODE
16 COLUMNS
CAPACITIVE KEYBOARD MATRIX

Figure 2

Keyboard Mode Register

Operating modes are selected by programming the keyboard mode register (KMR), whose format is illustrated in figure 3.

Bit KMR7 is used for testing the device. For normal operation, this bit should always be written to a 0.

Bits KMR6-KMR5 select the rollover modes for keyboard processing:

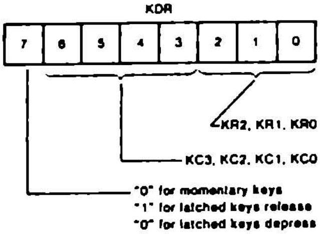
N-key Rollover: In this mode, the code corresponding to each key depression is loaded into the KDR as soon as that key is debounced, independent of the release of other keys. Two or more closures occurring within one scan cycle are considered to be simultaneous, which will set keyboard error in the keyboard status register (KSR1). As soon as the keyboard holding register is empty, the code in the KDR is transferred to the KHR and the KRDY status bit is set (KSR0).

N-Key Rollover With Latched Keys: This mode is the same as regular N-key rollover, except that the keys which are assigned to row 0 of the keyboard matrix (KR2-KR0 = 000) produce a code both when depressed and when released. The codes are independent of the states of the inputs at $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$. If one or more of the latched keys are depressed when the keyboard is enabled (after a keyboard reset), the corresponding codes will be sent out as the keys are scanned and debounced. Note that simultaneous latched keys will not set KERR (KSR1) and that latched keys will not be auto-repeat and will not be affected by the $\overline{\text{REPEAT}}$ input.

Two-Key Rollover: The first key code is loaded into the KDR immediately and the second code is loaded only after the first key is released. Simultaneous keys will set KERR (KSR1). If three or more keys remain closed at any given time, the KERR bit will also be set. All keys must then be released before the next KRET will be processed.

Two-Key Inhibit: All keys must be released between keystrokes; otherwise, KERR (KSR1) will be set.

Bit KMR4 specifies the key encoding mode. Each key is assigned four 8-bit codes, corresponding to the states of the $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs. If the encoded mode is programmed, the row/column address of the detected key is used to load one of the four key codes into the KDR. See table 2 for key code assignments. If the non-encoded mode is programmed, the row/column address is loaded directly into the KDR with the following format:


KDR
7
6
5
4
3
2
1
0
KR2, KR1, KR0
KC3, KC2, KC1, KC0
"0" for momentary keys
"1" for latched keys release
"0" for latched keys depress

COLUMN (KC3-KC0)	ROW (KR2-KR0)															
	0		1		2		3		4		5		6		7	
0	E0		C0		1B	ESC	09	HT	1F	US	1A	SUB	30	0	2B	+
	F0		D0		1B	ESC	09	HT	1F	US	1A	SUB	30	0	3B	;
	E0		C0		1B	ESC	09	• HT	1F	US	5A	Z	30	0	2B	+
	F0		D0		1B	ESC	09	• HT	1F	US	7A	z	30	0	3B	;
1	E1		C1		21	!	11	DC1	01	SOH	18	CAN	3D	=	2A	*
	F1		D1		31	1	11	DC1	01	SOH	18	CAN	2D	—	3A	:
	E1		C1		21	!	51	Q	41	A	58	X	3D	• =	2A	*
	F1		D1		31	1	71	q	61	a	78	x	2D	• —	3A	:
2	E2		C2		22	"	17	ETB	13	DC3	03	ETX	1E	RS	1F	US
	F2		D2		32	2	17	ETB	13	DC3	03	ETX	1E	RS	1F	US
	E2		C2		22	"	57	W	53	S	43	C	7E	~	7F	• DEL
	F2		D2		32	2	77	w	73	s	63	c	5E	↑	5F	• —
3	E3		C3		23	#	05	ENQ	04	EOT	16	SYN	1C	FS	1B	ESC
	F3		D3		33	3	05	ENQ	04	EOT	16	SYN	1C	FS	1B	ESC
	E3		C3		23	#	45	E	44	D	56	V	7C	¦	7B	{
	F3		D3		33	3	65	e	64	d	76	v	5C	\	5B	[
4	E4		C4		24	$	12	DC2	06	ACK	02	STX	08	BS	1D	GS
	F4		D4		34	4	12	DC2	06	ACK	02	STX	08	BS	1D	GS
	E4		C4		24	$	52	R	46	F	42	B	08	• BS	7D	}
	F4		D4		34	4	72	r	66	f	62	b	08	• BS	5D	]
5	E5		C5		25	%	14	DC4	07	BEL	0E	SO	10	DLE	08	BS
	F5		D5		35	5	14	DC4	07	BEL	0E	SO	10	DLE	08	BS
	E5		C5		25	%	54	T	47	G	4E	N	50	P	08	BS
	F5		D5		35	5	74	t	67	g	6E	n	70	p	08	• BS
6	E6		C6		26	&	19	EM	08	BS	0D	CR	00	NUL	09	HT
	F6		D6		36	6	19	EM	08	BS	0D	CR	00	NUL	09	HT
	E6		C6		26	&	59	Y	48	H	4D	M	60	`	09	• HT
	F6		D6		36	6	79	y	68	h	6D	m	40	@	09	• HT
7	E7		C7		27	'	15	NAK	0A	LF	3C	<	7F	DEL	20	SP
	F7		D7		37	7	15	NAK	0A	LF	2C	,	7F	DEL	20	SP
	E7		C7		27	'	55	U	4A	J	3C	<	7F	DEL	20	• SP
	F7		D7		37	7	75	u	6A	j	2C	,	7F	DEL	20	• SP
8	E8		C8		28	(	09	HT	0B	VT	3E	>	0A	LF	0B	VT
	F8		D8		38	8	09	HT	0B	VT	2E	.	0A	LF	0B	VT
	E8		C8		28	(	49	I	4B	K	3E	>	0A	LF	0B	• VT
	F8		D8		38	8	69	i	6B	k	2E	.	0A	LF	0B	• VT
9	E9		C9		29	)	0F	SI	0C	FF	3F	?	0D	CR	0A	LF
	F9		D9		39	9	0F	SI	0C	FF	2F	/	0D	CR	0A	LF
	E9		C9		29	)	4F	O	4C	L	3F	?	0D	CR	0A	• LF
	F9		D9		39	9	6F	o	6C	l	2F	/	0D	CR	0A	• LF
A	EA		CA		37	7	34	4	31	1	30	0	A0		A6	
	FA		DA		37	7	34	4	31	1	30	0	B0		B6	
	EA		CA		37	7	34	4	31	1	30	0	A0		A6	
	FA		DA		37	7	34	4	31	1	30	0	B0		B6	
B	EB		CB		38	8	35	5	32	2	2E	.	A1		A7	
	FB		DB		38	8	35	5	32	2	2E	.	B1		B7	
	EB		CB		38	8	35	5	32	2	2E	.	A1		A7	
	FB		DB		38	8	35	5	32	2	2E	.	B1		B7	
C	EC		CC		39	9	36	6	33	3	8F		A2		A8	
	FC		DC		39	9	36	6	33	3	8F		B2		B8	
	EC		CC		39	9	36	6	33	3	8F		A2		A8	
	FC		DC		39	9	36	6	33	3	8F		B2		B8	
D	ED		CD		90		93		82		95		A3		A9	
	FD		DD		90		93		82		95		B3		B9	
	ED		CD		90		93		82	•	95		A3	•	A9	•
	FD		DD		90		93		82	•	95		B3	•	B9	•
E	EE		CE		91		80		84		81		A4		AA	
	FE		DE		91		80		84		81		B4		BA	
	EE		CE		91		80	•	84		81	•	A4	•	AA	•
	FE		DE		91		80	•	84		81	•	B4	•	BA	•
F	EF		CF		92		94		83		96		A5		AB	
	FF		DF		92		94		83		96		B5		BB	
	EF		CF		92		94		83	•	96		A5	•	AB	•
	FF		DF		92		94		83	•	96		B5	•	BB	•

This row contains the latched keys when that mode is selected (KMR6, KMR5 = 00).

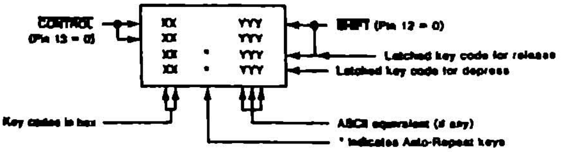
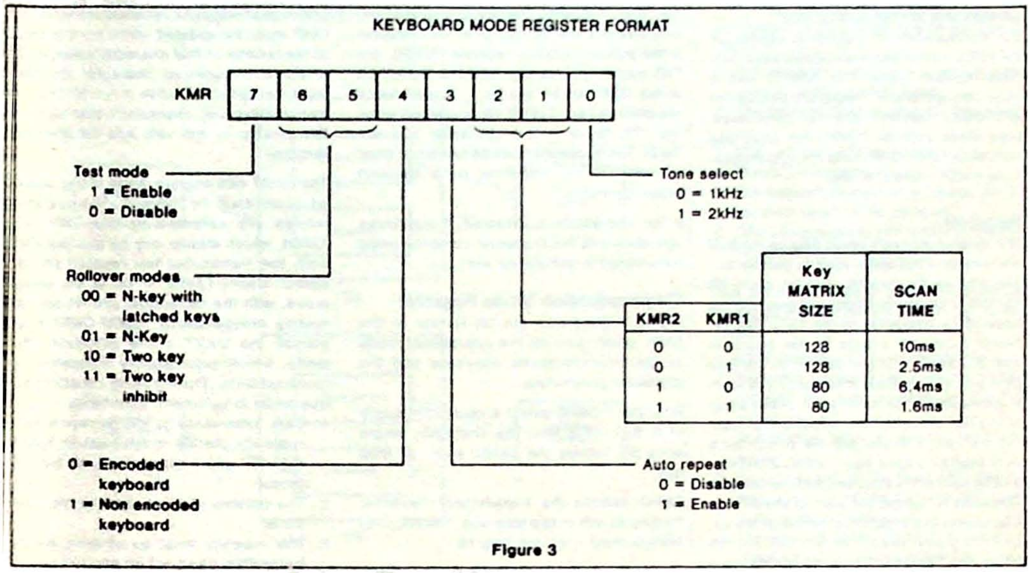

CONTROL (Pin 13 = 0)
XX YYY
XX YYY
XX • YYY
XX • YYY
SHIFT (Pin 12 = 0)
Latched key code for release
Latched key code for depress
Key codes in hex
ASCII equivalent (if any)
* indicates Auto-Repeat keys

TABLE 2. Standard Key Codes (HEX)

Preliminary


KEYBOARD MODE REGISTER FORMAT
KMR
7
6
5
4
3
2
1
0
Test mode
1 = Enable
0 = Disable
Rollover modes
00 = N-key with latched keys
01 = N-Key
10 = Two key
11 = Two key inhibit
0 = Encoded keyboard
1 = Non encoded keyboard
Tone select
0 = 1kHz
1 = 2kHz
Auto repeat
0 = Disable
1 = Enable

KMR2	KMR1	Key MATRIX SIZE	SCAN TIME
0	0	128	10ms
0	1	128	2.5ms
1	0	80	6.4ms
1	1	80	1.6ms

Figure 3

Bit KMR3 enables the auto-repeat mode. In this mode, if a key that is programmed for auto-repeat is depressed for longer than one-half second, the key code will be loaded into the KDR approximately 15 times per second until that key is released. Only the non-control key codes will auto-repeat, i.e. $\overline{\text{CONTROL}}$ = 1 Table 2 specifies the auto-repeat keys.

KMR2 and KMR1 select the key matrix size and debounce time (scan rate). The keyboard row outputs (KR2, KR1, KR0) always scan from 0 to 7. The column outputs (KC3, KC2, KC1, KC0) scan from 0 to 15 for a 128 key matrix and from 0 to 9 for an 80 key matrix.

KMR0 selects between a 1kHz and 2kHz frequency to be output on the TONE pin in response to a ring tone command.

Keyboard Status Register

The keyboard status register (KSR) provides operational feedback to the CPU. Its format is illustrated in figure 4.

KSR7, 6 and 4 reflect the state of the inputs at the corresponding pins. $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are latched at the time the key is accepted. As the verified codes are loaded into the KDR, the corresponding states of $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are loaded into the KSR. $\overline{\text{REPEAT}}$ is updated on every matrix sample. The status bits are the complements of the input levels.

KSR5 reflects the state of the internal shift lock flag which is controlled by the set/reset shift lock commands.

KSR3 indicates that the keyboard controller is enabled. It is controlled by the set/clear keyboard enable command.

Keyboard overrun (KSR2) is set when both the KHR and KDR are full and a third key is validated. The original content of the KHR is preserved and the content of the KDR is overwritten with the new key code. This bit can be specified (by IMR1) to generate an interrupt and is cleared by the reset command with D2 = 1.

Keyboard error (KSR1) is set when the operator depresses more keys than are allowed in the selected rollover mode, or when keys are depressed simultaneously (within one scan cycle). This bit can be specified (by IMR3) to generate an interrupt and is cleared by the reset command with D1 = 1.

Keyboard data ready (KSR0) is set when the key code or address is transferred from the KDR to the KHR. This bit can be specified (by IMR2) to generate an interrupt. It is cleared when the CPU reads the KHR.

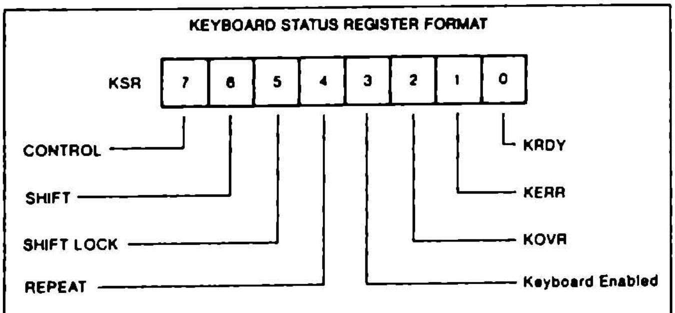

KEYBOARD STATUS REGISTER FORMAT
KSR
7
6
5
4
3
2
1
0
CONTROL
SHIFT
SHIFT LOCK
REPEAT
KRDY
KERR
KOVR
Keyboard Enabled

Figure 4

Communications Controller

The communications controller section of the PKCC comprises a full duplex asynchronous receiver/transmitter (UART) with a baud rate generator. Registers associated with these elements are the communications mode register (CMR), the baud rate control register (BRR), and the communications status register (CSR).

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again after a delay of one half of the bit time. If RxD is then high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RxHR) and the RxRDY bit in the CSR is set to a 1. If the character length is less than eight bits, the most significant unused bits in the RxHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the space is interpreted as a start bit.

The parity error, framing error and overrun error (if any) are strobed into the CSR at the received character boundary. If a break condition is detected (RxD is low for the entire character including the stop bit) only one character consisting of all zeros will be transferred to the RxHR and the received break bit in the CSR is set to 1 (RxRDY is not set when a break is received). The RxD input must return to a high condition for one bit time before a search for the next start bit begins.

Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the transmit holding register (TxHR), the TxD output remains high and the TxEMT bit in the CSR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxHR. The transmitter can be forced to send a continous low condition by a transmit break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out.

Communication Mode Register

Figure 5 illustrates the bit format of the CMR, which controls the operational mode of the communications controller and the character parameters.

Bits CMR1-CMR0 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity, start, or stop bits.

CMR2 selects the transmitted character framing as one or two stop bits. The receiver always checks for one stop bit.

The parity format is selected by bits CMR4 and CMR3. If parity or force parity is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. CMR5 selects odd or even parity and determines the polarity of the parity bit in the force parity mode.

The bits in the mode register affecting character assembly and disassembly (CMR5-CMR0) can be changed dynamically and affect the characters currently being assembled in RxSR and transmitted by TxSR. To affect assembly of a received character, the CMR must be updated within n − 1 bit times of the receipt of that character's start bit. To affect a transmitted character, the CMR must be updated within n − 1 bit times of transmitting that character's start bit. (n = the smaller of the new and old character lengths).

The UART can operate in one of four modes, as illustrated in figure 6. The operating modes are selected by bits CMR7 and CMR6, which should only be changed when both the transmitter and receiver are disabled. CMR7-CMR6 = 00 is the normal mode, with the transmitter and receiver operating independently. CMR7-CMR6 = 01 places the UART in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. Status bit TxRDY is not set. TxEMT operates normally.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Only the first character of a break condition is echoed; the TxD output will go high until the next received character is assembled.
7. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

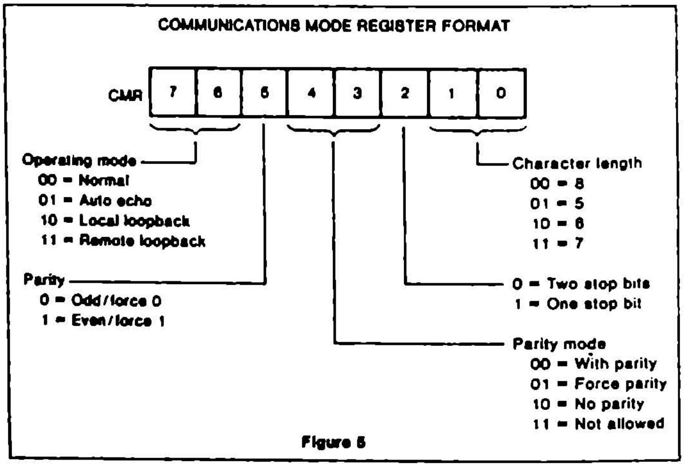
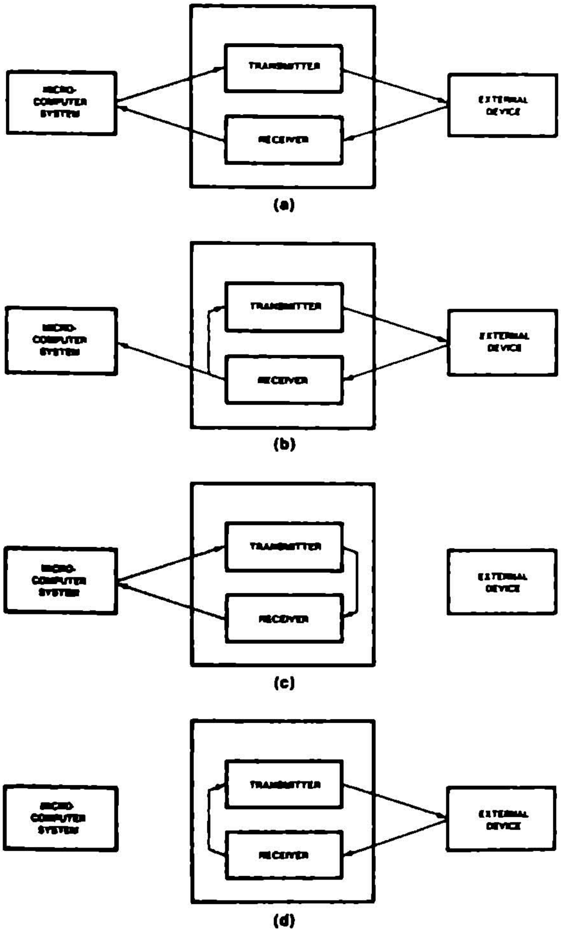


Figure 5

OPERATING MODES OF THE 2671 UART



(a) Normal operating mode.
(b) Automatic echo mode.
(c) Local loopback mode.
(d) Remote loopback mode.

Figure 6

Two diagnostic modes can also be configured. In local loopback mode (CMR7-CMR6 = 10):

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode (CMR7-CMR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. No data is sent to the local CPU, but the error status conditions (parity and framing) are set if required.
4. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.

Baud Rate Control Register

The baud rate control register (BRR) controls the frequency generated by the baud rate generator (BRG) and the clock source used by the receiver and transmitter. Its format is illustrated in figure 7.

BRR3-BRR0 select one of sixteen frequencies to be generated by the BRG. See table 3.

BRR7 and BRR6 select the source of the transmit and receive clocks. If external clocks are chosen, (BRR7 = 0 or BRR6 = 0), then the clock rate factor is determined by BRR5 and BRR4. The external clock input(s) should be the desired baud rate multiplied by the clock rate factor.

If internal clock(s) are specified, (BRR7 = 1 or BRR6 = 1), the clock is supplied by the internal baud rate generator at the selected baud rate. The clock rate factor for internally generated clocks is always 16. Pins 35 and 34 become outputs for transmit or receive clocks, respectively. See table 4 for the description and selection of these outputs.

BAUD RATE CONTROL REGISTER FORMAT

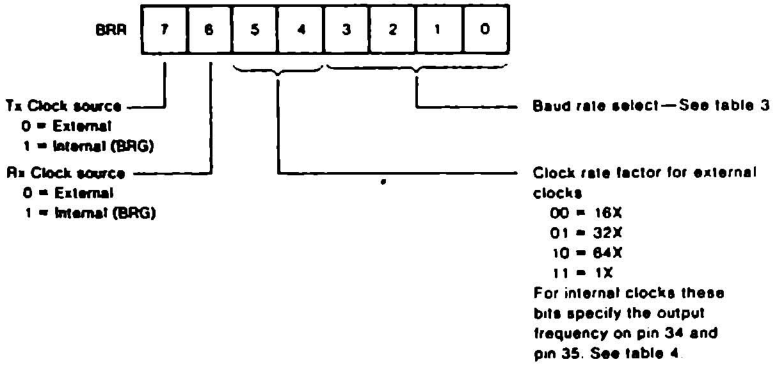

BRR
7
6
5
4
3
2
1
0
Tx Clock source
0 = External
1 = Internal (BRG)
Rx Clock source
0 = External
1 = Internal (BRG)
Baud rate select—See table 3
Clock rate factor for external clocks
00 = 16X
01 = 32X
10 = 64X
11 = 1X
For internal clocks these bits specify the output frequency on pin 34 and pin 35. See table 4

Figure 7

BRR3-0	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8 kHz	—	6144
0001	110	1.7598	−0.01	2793
0010	134.5	2.152	—	2284
0011	150	2.4	—	2048
0100	200	3.2	—	1536
0101	300	4.8	—	1024
0110	600	9.6	—	512
0111	1050	16.8329	+0.20	292
1000	1200	19.2	—	256
1001	1800	28.7438	−0.20	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	—	128
1100	4800	76.8	—	64
1101	9600	153.6	—	32
1110	19200	307.2	—	16
1111	38400	614.4	—	8

Table 3. BAUD RATE GENERATOR CHARACTERISTICS
(BRCLK = 4.9152MHz)

Preliminary

BRR7-BRR4	CLOCK SOURCE TxC	CLOCK SOURCE RxC	PIN FUNCTIONS PIN 34	PIN FUNCTIONS PIN 35	BRR3-BRR0 BAUD RATE SELECTION
00**	E	E	TxC	RxC	The baud rates are listed in table 3.
01**	E	I	TxC	1X	
10**	I	E	16X	RxC	
1100	I	I	1X	1X	
1101	I	I	1X	16X	
1110	I	I	16X	1X	
1111	I	I	16X	16X	

NOTES

1. ** = Clock rate factor for external clocks: 00 = 16X
   01 = 32X
   10 = 64X
   11 = 1X
2. E = External clock.
3. I = Internal clock (BRG).
4. 1X and 16X are clock outputs at 1 or 16 times the actual baud rate. For receive, the 1X output is the actual data sample clock.
5. BRR7-BRR6 = 01 or 10 not permitted in automatic echo or remote loopback modes unless BRR5-BRR4 = 00.

Table 4. BAUD RATE CONTROL REGISTER

Communications Status Register

Figure 8 illustrates the bit format of the communications status register (CSR), which provides UART status to the CPU.

Receiver ready (CSR0) indicates that a received character is assembled and transferred to the RxHR and is ready to be read by the CPU. This bit can be specified (by IMR0) to generate an interrupt and is reset by reading the RxHR.

Transmitter ready (CSR1) indicates that the TxHR is empty and ready to be loaded with a character. This bit will be cleared when the TxHR is loaded and has not yet transferred the character to the transmit shift register (TxSR). TxRDY is reset when the transmitter is disabled. It will be set when the transmitter is enabled, provided that no data was loaded into the TxHR during the time the transmitter was disabled. This bit can be specified (by IMR7) to generate an interrupt.

Transmitter empty (CSR2) indicates that the transmitter has underrun, i.e., both the TxHR and TxSR are empty. This bit can only be set after transmission of at least one character, and is cleared when the TxHR is loaded by the CPU. TxEMT is reset when the transmitter is disabled. This bit can be specified (by IMR6) to generate an interrupt.

CSR3 will be set when the PKCC receives a command to transmit a break. This bit will be cleared after the break is completed.

Received break (CSR4) indicates that an all zero character of the programmed length has been received without a stop bit. Breaks originating in the middle of a received character can be detected. This bit is cleared when RxD returns to a high state for at least one bit time.

Receiver overrun (CSR5) indicates that the previous character in the RxHR has not been read by the CPU and that a new character has been loaded into the RxHR. This bit is cleared by a reset command with D3 = 1.

Framing error (CSR6) indicates that the stop bit has not been detected. The stop bit check is made in the middle of the first stop bit position. This bit is cleared by a reset command with D3 = 1.

Parity error (CSR7) indicates that a character was received with incorrect parity when 'with parity' or 'force parity' is enabled. This bit is cleared by a reset command with D3 = 1.

Interrupt Controller

The 2671 contains a maskable interrupt status register (ISR) which can be enabled to generate an active low interrupt request on the $\overline{INTR}$ output. The eight interrupt conditions in the ISR are individually enabled by writing a 1 into the corresponding bit of the interrupt mask register (IMR).

Each of the interrupt conditions is assigned a priority and a vector. When an enabled ISR bit is set, the 2671 asserts the $\overline{INTR}$ output. If the CPU activates the $\overline{INTA}$ input, the 2671 responds by placing the corresponding 8-bit vector on the data bus (D7-D0). If multiple interrupts are pending, the vector corresponds to the condition with the highest priority. The interrupt will persist until all pending interrupt conditions are cleared.

The ISR can also be polled by reading at address A2-A0 = 000. All pending interrupt conditions which are enabled by the IMR will be read independent of priority.

COMMUNICATIONS STATUS REGISTER FORMAT

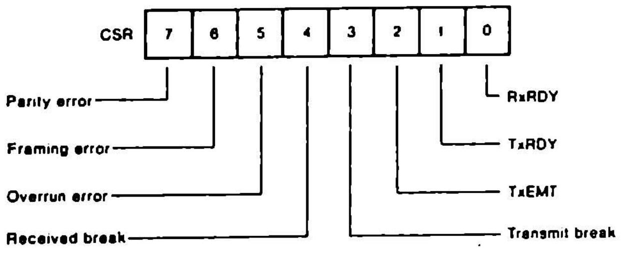


Figure 8

The bit assignments of the ISR and IMR and corresponding vectors and priorities are listed in table 5.

COMMANDS

In addition to the control exercised by programming of the PKCC control registers, several functions can be performed by executing command operations. There are two classes of commands which are initiated by writing to the 2671 at address A2-A0 = 000 (reset command) and address A2-A0 = 111 (miscellaneous commands). Individual commands are specified by the bit pattern on the data bus (D7-D0).

Reset Commands

The reset command bit format is illustrated in figure 9 and the detail command descriptions are given in table 6.

A reset command with D7-D0 = 111XXXX1 is a master reset for the 2671. This command must be given following a power on condition to release the internal power on reset latch which deactivates the 2671 on power up.

Miscellaneous Commands

The miscellaneous command format is illustrated in figure 10.

The transmit break commands force a break (steady low output) on the TxD pin immediately or after the character in the TxSR (if any) is transmitted. A timed break lasts for approximately 200ms, and a character break lasts for one character time including parity and stop bit time. In either case, TxRDY (CSR1) will be set at the beginning of the break which can be extended indefinitely (by 200ms or one character time increments) by reasserting the command in response to TxRDY. Note that these commands reset TxRDY. When a transmit break command is asserted, CSR3 will be set. This bit will be cleared after the break is completed.

The ring tone commands cause the tone generator to output a square wave on the TONE output. The tone durations are specified by the commands:

Ring tone short = 25ms
Ring tone long = 100ms

The tone frequency is either 1kHz or 2kHz, as specified by KMR0.

BIT IN IMR/ISR	INTERRUPT CONDITION	PRIORITY	VECTOR ON D7-D0 BINARY	VECTOR ON D7-D0 HEX	CONDITION RESET BY:
IMR0/ISR0	RxRDY	1	11001111	CF	Read RxHR
IMR1/ISR1	KOVR	2	11010111	D7	Reset CMD (D2 = 1)
IMR2/ISR2	KRDY	3	11011111	DF	Read KHR
IMR3/ISR3	KERR	4	11100111	E7	Reset CMD (D1 = 1)
IMR4/ISR4	XINT[1]	5	11101111	EF	External
IMR5/ISR5	ΔBREAK[2]	6	11110111	F7	Reset CMD (D4 = 1)
IMR6/ISR6	TxEMT	7	11000111	C7	Load TxHR
IMR7/ISR7	TxRDY	8	11000111	C7	Load TxHR

NOTES:

1. XINT is an input from an external interrupt source, active low (pin 21).
2. ΔBREAK refers to the change of a received break condition.

Table 5. INTERRUPT MASK REGISTER (IMR) AND INTERRUPT STATUS REGISTER (ISR)

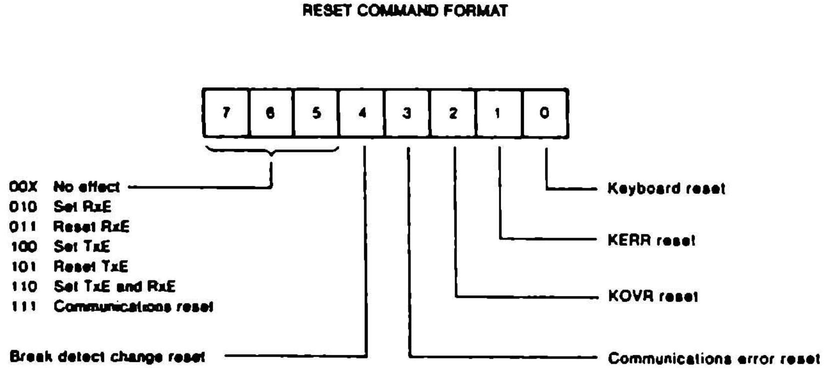


Figure 9

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

COMMAND	RESETS	COMMENTS
Keyboard reset	KMR7-KMR0 KSR5, KSR3-KSR0 IMR3-IMR1	The keyboard controller is reset, ignoring the input at KRET
KERR reset	KSR1	Keyboard error status bit reset
KOVR reset	KSR2	Keyboard overrun status bit reset
Communications error reset	CSR7-CSR5	Resets the receiver overrun, parity, and framing error status bits.
Break detect change reset	ISR5	Resets the break detect change bit in the interrupt status register.
Set RxE	See note.	Enables receiver operation.
Reset RxE	CSR7-CSR4, CSR0 See note.	Disables the receiver.
Set TxE	See note.	Enables transmitter operation.
Reset TxE	CSR3-CSR1 See note.	Disables the transmitter. Sets the TxD output to a 1 after transmitting the character in TxSR.
Communications reset	CMR, CSR, BRR, TxE, RxE, IMR7-IMR5, IMR0	Resets the communication controller. The RxD input is ignored and the TxD output is set to a 1.
Master reset	CMR, CSR, BRR, TxE, RxE, KMR, KSR5, KSR3-KSR0, IMR7-IMR0. Releases the internally latched power on reset	Resets the keyboard and communication controllers. Inputs at KRET and RxD are ignored and the TxD output is set to a 1.

NOTE
Command does not affect the CMR or the BRR.

Table 6. RESET COMMAND DESCRIPTION

The set/clear shift lock commands control the state of the internal shift lock flip flop. When shift lock is set, the keyboard controller encodes all key depressions as if the SHIFT input was asserted. The state of the shift lock flip flop is reflected in KSR5.

The set keyboard enable command enables the keyboard controller and sets KSR3 in the keyboard status register. The clear keyboard enable command resets KSR3 and disables key processing at the KRET input. The keyboard controller is not reset by this command, and the current state of the keyboard (key depressions and latched key states) is preserved internally. When the keyboard is subsequently enabled, key processing resumes, old and new keys are debounced, and latched keys are encoded if there has been a change in their state.

MASK PROGRAMMABLE OPTIONS

Characteristics of certain portions of the PKCC are internally programmed by means of a read only memory. The items which can be programmed are:

- Key codes
- Auto-repeat keys
- Scan times, tone frequency, and tone duration
- Baud rates
- Interrupt vectors

Consult your local Signetics representative for costs, minimum quantities, and data submission requirements for customized versions of the PKCC.

MISCELLANEOUS COMMANDS FORMAT

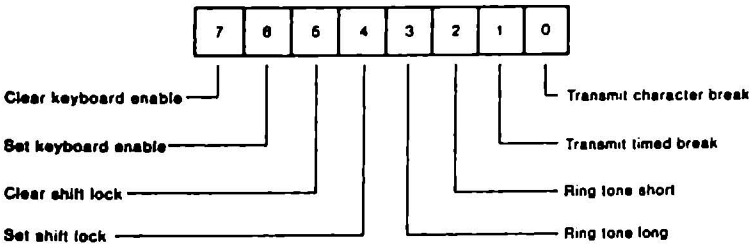


Figure 10

PROGRAMMABLE KEYBOARD & COMM CONTROLLER (PKCC) SC2671

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS Min	Typ	Max	UNIT
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage	Except XTAL 1 (see note 13)	2.0			V
V_{OL}	Output low voltage	$I_{OL} = 1.6mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$)	$I_{OH} = -100\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	TENTATIVE LIMITS Min	Typ	Max	UNIT
Read timing[7]						
t_{AS}	Address setup to $\overline{RD}$		50			ns
t_{CS}	$\overline{CE}$ setup to $\overline{RD}$		50			ns
t_{PW}	$\overline{RD}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{RD}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{RD}$		0			ns
t_{DD}	Data delay for read	$C_L = 150pF$			200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$	10		100	ns
t_{AD}	Access delay from any read to next read or write		250			ns
Write timing[8]						
t_{AS}	Address setup to $\overline{WR}$		50			ns
t_{CS}	$\overline{CE}$ setup to $\overline{WR}$		50			ns
t_{PW}	$\overline{WR}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{WR}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{WR}$		0			ns
t_{DS}	Data setup		100			ns
t_{DH}	Data hold		0			ns
t_{AD}	Access delay from any write to next read or write		250			ns
t_{AD}	Access delay from reset command to next read or write		1.0			μs

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 55°C/W junction to ambient (IWA ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum and time measurements are referenced at input voltages of 0.8V, 2.0V and at output voltages of 0.8V, 2.0V as appropriate, unless otherwise specified.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. See figure 11.
8. See figure 12.

AC ELECTRICAL CHARACTERISTICS (Cont.)

PARAMETER		TEST CONDITIONS	TENTATIVE LIMITS			UNIT
			Min	Typ	Max	
Interrupt acknowledge timing[9]						
t_{PW}	$\overline{\text{INTA}}$ pulse width		300			ns
t_{DDI}	Data delay time for interrupt vector	C_L = 150pF			250	ns
t_{DFI}	Data bus floating time after $\overline{\text{INTA}}$	C_L = 150pF	10		100	ns
t_{ADI}	$\overline{\text{INTA}}$ to $\overline{\text{INTA}}$ access delay		300			ns
$\overline{\text{INTR}}$ reset timing[10]						
t_{RI}	$\overline{\text{INTR}}$ delay from:					
	Read RxHR (RxRDY)				400	ns
	Read KHR (KRDY)				400	ns
	Reset commands (KOVR,KERR,BREAK)				450	ns
	Load TxHR (TxEMT,TxRDY)				400	ns
	Mask bit reset				300	ns
Keyboard timing[11]						
f_{KCLK}	KCLK frequency			409		kHz
t_{KBD}	KR_i, KC_i to KRET sample delay:					
	FAST SCAN		12.0			µs
	SLOW SCAN		55.0			µs
t_{POS}	Scan time per matrix position:					
	FAST SCAN			20		µs
	SLOW SCAN			80		µs
t_{KRD}	KDRES delay from KCLK	C_L = 150pF			400	ns
t_{KRH}	KDRES hold from KCLK	C_L = 150pF			400	ns
t_{HYSD}	$\overline{\text{HYS}}$ delay from KCLK	C_L = 150pF			600	ns
t_{RCD}	KR_i, KC_i delay from KCLK	C_L = 150pF			400	ns
UART timing[12]						
t_{RXS}	RxD setup time		200			ns
t_{RxH}	RxD hold time		200			ns
t_{TxD}	TxD delay from falling edge of TxC	C_L = 150pF			300	ns
t_{TCS}	Skew between TxD transition and falling edge of TxC output	C_L = 150pF		0		ns
t_{BRH}	XTAL1 clock high[13]		70			ns
t_{BRL}	XTAL1 clock low[13]		70			ns
f_{BRG}	BRG input frequency		1.0	4.9152	5.075	MHz
$f_{R/T}$	TxC or RxC input frequency	Clock rate factor = 16X, 32X, 64X			1.3	MHz
$f_{R/T}$	TxC or RxC input frequency	Clock rate factor = 1X			1.0	MHz
$t_{R/TH}$	TxC or RxC clock high		350			ns
$t_{R/TL}$	TxC or RxC clock low		350			ns

NOTES

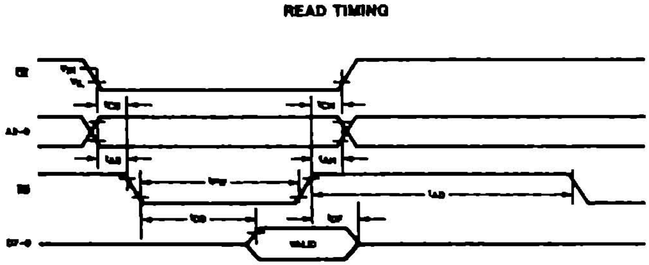
[9] See figure 13

[10] See figure 14

[11] See figure 15 and 16

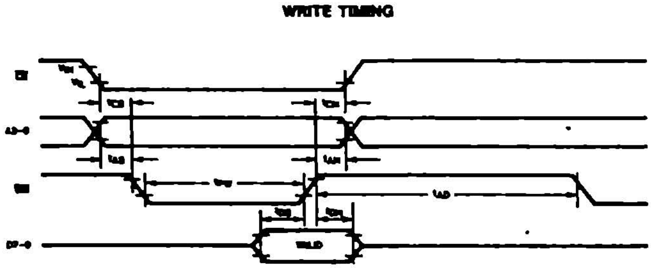
[12] See figure 17, 18, and 19

[13] See figures 20 and 21 for XTAL1, XTAL2 connections for driving XTAL2 with an external clock. Input levels for XTAL1 and XTAL2 are $V_{IL} \le 0.8V$, $V_{IH} \ge 4.0V$, and t_{BRL} and t_{BRH} are measured at these levels


READ TIMING

A3-0
D7-0
VALID

Figure 11


WRITE TIMING

A3-0
D7-0
VALID

Figure 12

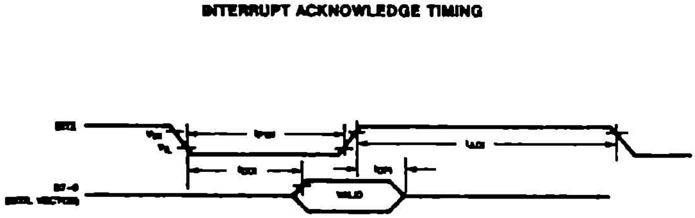

INTERRUPT ACKNOWLEDGE TIMING

Figure 13

INTERRUPT RESET TIMING

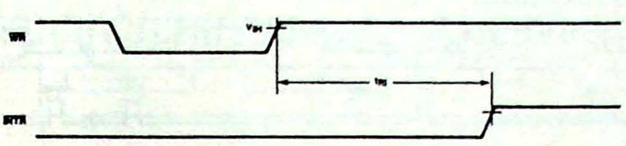
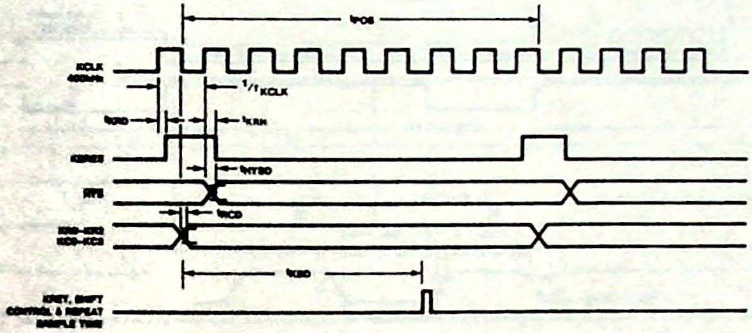

WR
VIH
tRI
INTR

Figure 14

KEYBOARD SCAN TIMING


tPOS
KCLK
400kHz
1/fKCLK
tKRD
tKRH
KBRES
tHYSD
HYS
tKCD
KR0-KR2
KC0-KC3
tKSD
KRET, SHIFT
CONTROL & REPEAT
SAMPLE TIME

NOTE
SCAN TIMING SHOWN IS FOR FAST SCAN (KMR1 = 1). FOR SLOW SCAN (KMR1 = 0) ALL SIGNALS EXCEPT KCLK RUN AT ¼ THE SHOWN RATES.

Figure 15

KEYBOARD TIMING

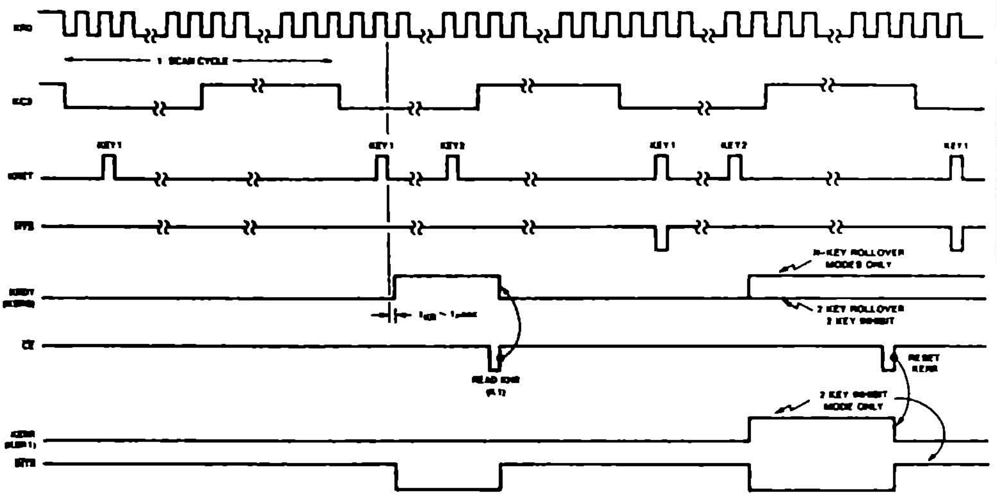
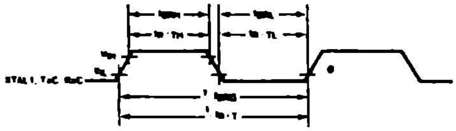
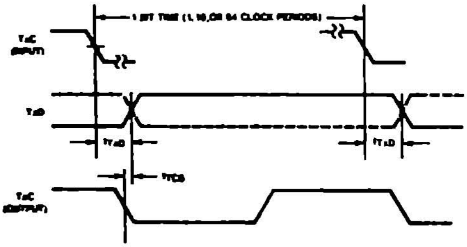

1 SCAN CYCLE
KEY 1
KEY 1
KEY 2
KEY 1
KEY 2
KEY 1
N-KEY ROLLOVER MODES ONLY
2 KEY ROLLOVER 2 KEY INHIBIT
RESET KERR
2 KEY INHIBIT MODE ONLY

Figure 16

CLOCK



TRANSMIT


TxC (INPUT)
TxD
TxC (OUTPUT)

RECEIVE

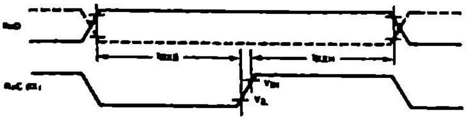


Figure 17

TRANSMITTER TIMING
(5-BIT CHARACTERS, NO PARITY, 2 STOP BITS)

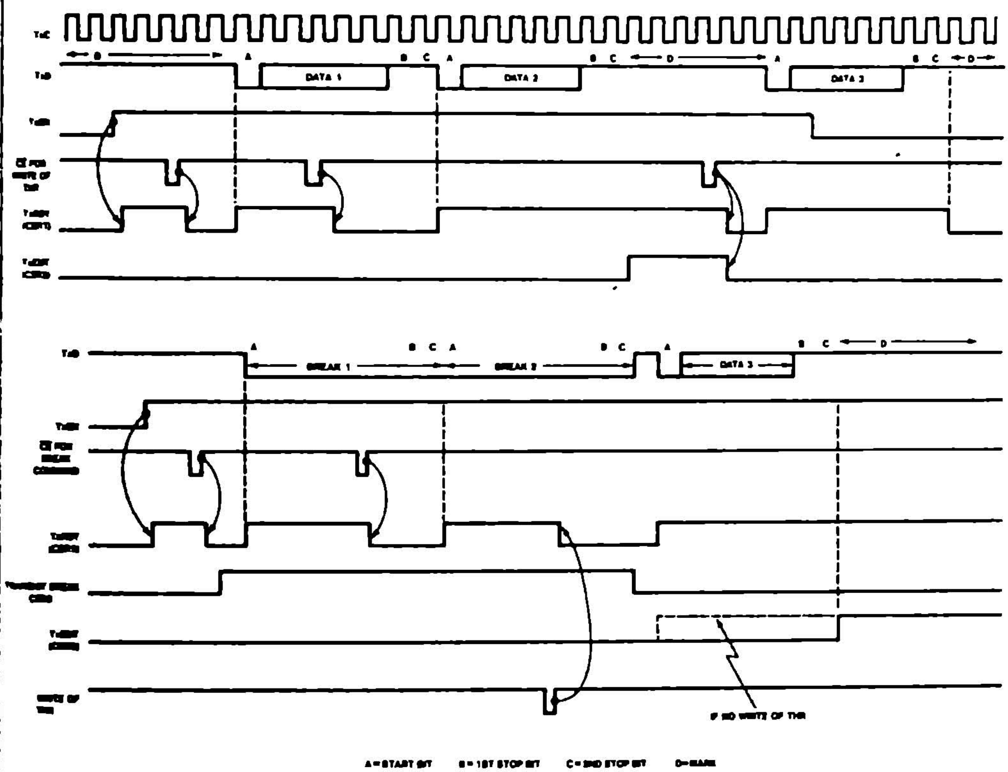

TxC
TxD
B
A
DATA 1
B C A
DATA 2
B C D
A
DATA 3
B C D
CE FOR WRITE OF THR
TxRDY
TxEMT
TxD
A
BREAK 1
B C A
BREAK 2
B C A
DATA 3
B C D
CE FOR BREAK COMMAND
TxRDY
TRANSMIT BREAK
TxEMT
WRITE OF THR
IF NO WRITE OF THR
A=START BIT
B=1ST STOP BIT
C=2ND STOP BIT
D=MARK

Figure 18

RECEIVER TIMING
(5-BIT CHARACTERS, NO PARITY, 2 STOP BITS)

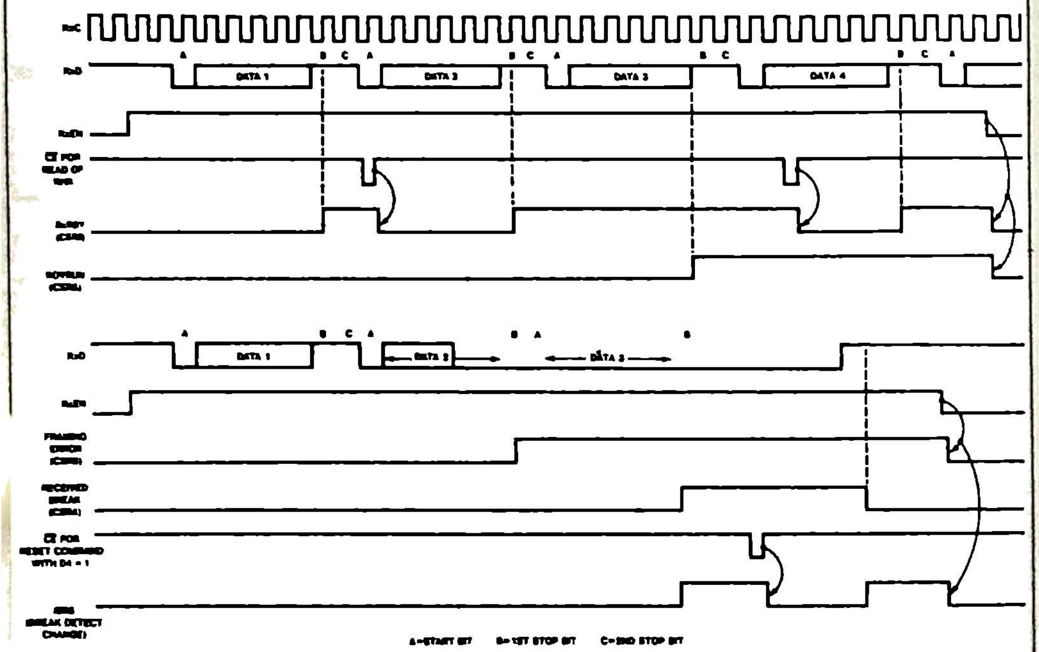

RxC
RxD
A
B
C
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
CE FOR READ OF RHR
RxRDY (CSR0)
ROVRUN (CSR1)
FRAMING ERROR (CSR2)
RECEIVED BREAK (CSR4)
CE FOR RESET COMMAND WITH D4 = 1
BDC (BREAK DETECT CHANGE)
A=START BIT
B=1ST STOP BIT
C=2ND STOP BIT

Figure 19

CRYSTAL CONNECTIONS BRG CLOCK

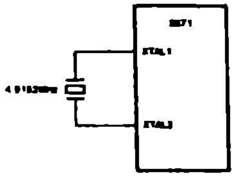

2671
XTAL 1
XTAL 2
4.9152MHz

Figure 20

CONNECTION FOR EXTERNAL BRG CLOCK SOURCE

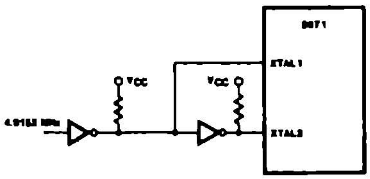

2671
XTAL 1
XTAL 2
VCC
VCC
4.9152 MHz

Figure 21

	7	6	5	4	3	2	1	0
	Test mode	Rollover modes		Keyboard	Auto repeat			Tone select
KMR	1 = Enable 0 = Disable	00 = N-key with latched keys 01 = N-key 10 = Two keys 11 = Two key inhibit		0 = Encoded 1 = Non encoded	0 = Disable 1 = Enable	KMR2 KMR1 Key Matrix Size Scan Time 0 0 128 10ms 0 1 128 2.6ms 1 0 80 6.4ms 1 1 80 1.6ms		0 = 1kHz 1 = 2kHz
KSR	CONTROL	SHIFT	SHIFT LOCK	REPEAT	Keyboard Enabled	KOVR	KERR	KRDY
	Operating Mode		Parity	Parity Mode		Stop Bits	Character Length	
CMR	00 = Normal 01 = Auto echo 10 = Local loopback 11 = Remote loopback		0 = Odd/ force 0 1 = Even/ force 1	00 = With parity 01 = Force parity 10 = No parity 11 = Not allowed		0 = Two 1 = One	00 = 8 01 = 5 10 = 6 11 = 7	
	Tx Clock source	Rx Clock source	Clock rate factor for external clocks		Baud rate select (BRR3 - BRR0 in hex)			
BRR	0 = External 1 = Internal (BRG)	0 = External 1 = Internal (BRG)	00 = 16X 01 = 32X 10 = 64X 11 = 1X For internal clocks these bits specify the output frequency on pins 34 and 35 (table 4).		0 = 50 1 = 110 2 = 134.5 3 = 150	4 = 200 5 = 300 6 = 600 7 = 1050	8 = 1200 9 = 1800 A = 2000 B = 2400	C = 4800 D = 9600 E = 19200 F = 38400
					(BRCLK = 4.9152MHz)			
CSR	Parity error	Framing error	Overrun error	Received break	Transmit break	TxEMT	TxRDY	RxRDY
IMR/ISR	TxRDY	TxEMT	BREAK CHANGE	XINT	KERR	KRDY	KOVR	RxRDY
Reset Command Format	00X = No effect 010 = Set RxE 011 = Reset RxE 100 = Set TxE 101 = Reset TxE 110 = Set TxE and RxE 111 = Communications reset			Break detect change reset	Communications error reset	KOVR reset	KERR reset	Keyboard reset
Miscellaneous Commands Format	Clear keyboard enable	Set keyboard enable	Clear shift lock	Set shift lock	Ring tone long	Ring tone short	Transmit timed break	Transmit character break

Table 7. Register Format Summary

JANUARY 1982

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC)

SC2672

Preliminary

DESCRIPTION

The Signetics 2672 Programmable Video Timing Controller (PVTC) is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The PVTC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the PVTC.

A minimum CRT terminal system configuration consists of a PVTC, a 2671 Keyboard and Communication Controller (PKCC), a 2670 Display Character and Graphics Generator (DCGG), a 2673 Video and Attributes Controller (VAC), a single chip microcomputer such as the 8048, a display buffer RAM, and a small amount of TTL for miscellaneous address decoding, interface, and control. Typically, the package count for a minimum system is between 15 and 20 devices; system complexity can be enhanced by upgrading the microprocessor and expanding via the system address and data busses.

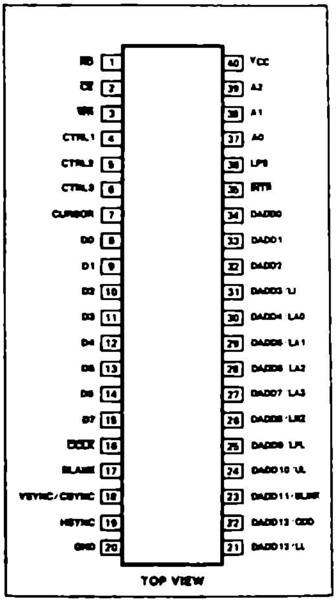
FEATURES

- 4MHz character rate
- Up to 256 characters per row
- 1 to 16 raster lines per character row
- Up to 128 character rows per frame
- Programmable horizontal and vertical sync generators
- Interlaced or non−interlaced operation
- Up to 16K RAM addressing for multiple page operation
- Automatic wraparound of RAM
- Addressable incrementable and readable cursor
- Programmable cursor size, position, and blink
- Split screen and horizontal scroll capability
- Light pen register
- Selectable buffer interface modes
- Dynamic RAM refresh
- Completely TTL compatible
- Single +5 volt power supply
- Power on reset circuit

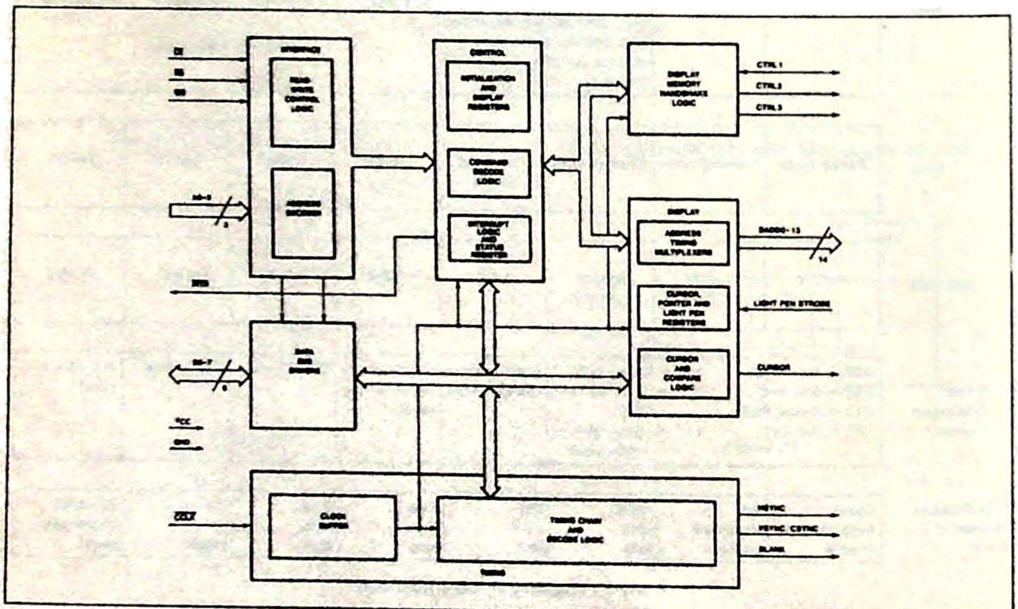
APPLICATIONS

- CRT terminals
- Word processing systems
- Small business computers
- Home Computers

PIN CONFIGURATION


RD 1
CE 2
WR 3
CTRL1 4
CTRL2 5
CTRL3 6
CURSOR 7
D0 8
D1 9
D2 10
D3 11
D4 12
D5 13
D6 14
D7 15
DCLK 16
BLANK 17
VSYNC/CSYNC 18
HSYNC 19
GND 20
40 VCC
39 A2
38 A1
37 A0
36 LPS
35 INTR
34 DADD0
33 DADD1
32 DADD2
31 DADD3/LI
30 DADD4/LA0
29 DADD5/LA1
28 DADD6/LA2
27 DADD7/LA3
26 DADD8/LR2
25 DADD9/LRL
24 DADD10/UL
23 DADD11/BLINK
22 DADD12/ODD
21 DADD13/LL
TOP VIEW

BLOCK DIAGRAM


CE
RD
WR
A0-2
3
INTR
D0-7
8
VCC
GND
DCLK
INTERFACE
READ/WRITE CONTROL LOGIC
ADDRESS DECODER
DATA BUS DRIVERS
CONTROL
INITIALIZATION AND DISPLAY REGISTERS
COMMAND DECODE LOGIC
INTERRUPT LOGIC AND STATUS REGISTER
DISPLAY MEMORY HANDSHAKE LOGIC
CTRL1
CTRL2
CTRL3
DISPLAY
ADDRESS TIMING MULTIPLEXERS
DADD0-13
14
CURSOR, POINTER AND LIGHT PEN REGISTERS
LIGHT PEN STROBE
CURSOR AND COMPARE LOGIC
CURSOR
CLOCK BUFFER
TIMING CHAIN AND DECODE LOGIC
TIMING
HSYNC
VSYNC/CSYNC
BLANK

VALVO

JANUARY 1982

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SC2672

Preliminary

ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ± 5%, T_A = 0°C to 70°C
Ceramic DIP	SC2672C4I40
Plastic DIP	SC2672C4N40

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
A0–A2	37–39	I	**Address Lines:** Used to select PVTC internal registers for read/write operations and for commands.
D0–D7	8–15	I/O	**8-Bit Bidirectional Three-State Data Bus.** Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the PVTC takeplace over this bus. The direction of the transfer is controlled by the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ inputs when the $\overline{\text{CE}}$ input is low. When the $\overline{\text{CE}}$ input is high, the data bus is in the three-state condition.
$\overline{\text{RD}}$	1	I	**Read Strobe:** Active low input. A low on this pin while $\overline{\text{CE}}$ is low causes the contents of the register selected by A0–A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of $\overline{\text{RD}}$
$\overline{\text{WR}}$	3	I	**Write Strobe:** Active low input. A low on this pin while $\overline{\text{CE}}$ is also low causes the contents of the data bus to be transferred to the register selected by A0–A2. The transfer occurs on the trailing (rising) edge of $\overline{\text{WR}}$.
$\overline{\text{CE}}$	2	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the PVTC are enabled on D0–D7 as controlled by the $\overline{\text{WR}}$, $\overline{\text{RD}}$, and A0–A2 inputs. When $\overline{\text{CE}}$ is high, the PVTC is effectively isolated from the data bus and D0–D7 are placed in the three-state condition.
$\overline{\text{CCLK}}$	16	I	**Character Clock:** Timing signal derived from the video dot clock which is used to synchronize the PVTC's timing functions.
HSYNC	19	O	**Horizontal Sync:** Active high output which provides video horizontal sync pulses. The timing parameters are programmable.
VSYNC/CSYNC	18	O	**Vertical Sync/Composite Sync:** A control bit selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.
BLANK	17	O	**Blank:** This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on DADD3 thru DADD13 are valid on the trailing edge of BLANK.
CURSOR	7	O	**Cursor Gate:** This active high output becomes active for a specified number of scan lines when the address contained in the cursor registers match the address output on DADD0 thru DADD13. The first and last lines of the cursor and a blink option are programmable.
$\overline{\text{INTR}}$	35	O	**Interrupt Request:** Open drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after power on reset or a master reset command.
LPS	36	I	**Light Pen Strobe:** Positive edge triggered input indicating a light pen hit. Causes the current value of the display address to be strobed into the light pen register.
CTRL1	4	I/O	**Handshake Control 1:** In independent mode, provides an active low write data buffer ($\overline{\text{WDB}}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{\text{PBREQ}}$) input which indicates that the CPU desires to access the display memory. This control output has no meaning in row buffer mode.
CTRL2	5	O	**Handshake Control 2:** In independent mode, provides an active low read data buffer ($\overline{\text{RDB}}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, this is an active low bus external enable ($\overline{\text{BEXT}}$) output which indicates that the PVTC has relinquished control of the display memory (DADD0–DADD13 are in the three-state condition) in response to a CPU bus request. $\overline{\text{BEXT}}$ also goes low in response to a 'display off and float DADD' command. In row buffer mode, it is an active low bus request ($\overline{\text{BREQ}}$) output which halts the CPU during a line DMA.

PIN DESIGNATION (cont.)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
CTRL3	6	O	**Handshake Control 3:** In independent mode, provides the active low buffer chip enable ($\overline{\text{BCE}}$) signal to the display memory. In transparent and shared modes, provides an active low bus acknowledge ($\overline{\text{BACK}}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, this is an active high memory bus control (MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.
DADD0–DADD13	34–21	O	**Display Address:** Used by the PVTC to address up to 16K of display memory. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD3 thru DADD13 and are valid at the trailing edge of BLANK. These control signals are: **DADD3/LI** **Line Interlace:** Replaces DADD4/LA0 as the least significant line address for interlaced sync and video applications. A low indicates an even row of an even field or an odd row of an odd field. **DADD4–DADD7/LA0–LA3** **Line Address:** Provides the number of the current scan line within each character row. **DADD8/LNZ** **Line Zero:** Asserted before the first scan line in each character row. **DADD9/LPL** **Light Pen Line:** Asserted before the scan line which matches the programmed light pen line position (line 3, 5, 7, or 9). **DADD10/UL** **Underline:** Asserted before the scan line which matches the programmed underline position (line 0 thru 15). **DADD11/BLINK** **Blink frequency:** Provides an output divided down from the vertical sync rate. **DADD12/ODD** **Odd Field:** Active high signal which is asserted before each scan line of the odd field when interlace is specified. **DADD13/LL** **Last Line:** Asserted before the last scan line of each character row.
V_{CC}	40	I	**Power Supply:** +5 volts ± 5% power input.
GND	20	I	**Ground:** Signal and power ground input.

FUNCTIONAL DESCRIPTION

As shown on the block diagram, the PVTC contains the following major blocks:

- Data bus buffer
- Interface Logic
- Operation Control
- Timing
- Display Control
- Buffer Control

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the PVTC.

Interface Logic

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are as shown in table 1.

Table 1 PVTC ADDRESSING

A2	A1	A0	READ (RD=0)	WRITE (WR=0)
0	0	0	Interrupt register	Initialization registers[1]
0	0	1	Status register	Command register
0	1	0	Screen start address lower register	Screen start address lower reg.
0	1	1	Screen start address upper register	Screen start address upper reg.
1	0	0	Cursor address lower register	Cursor address lower register
1	0	1	Cursor address upper register	Cursor address upper register
1	1	0	Light pen address lower register	Display pointer address lower reg.
1	1	1	Light pen address upper register	Display pointer address upper reg.

NOTE

[1] There are 11 initialization registers which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split screen register) is accessed. The pointer then continues to point to the split screen register. Upon power-up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the "load IR address pointer" command.

JANUARY 1982

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SC2672

Preliminary

Operation Control

The operation control section decodes configuration and operation commands from the CPU and generates appropriate signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating mode, the interrupt logic, and the status register which provides operational feedback to the CPU.

Timing

The timing section contains the counters and decoding logic necessary to generate the monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

Display Control

The display control section generates linear addressing for up to 16K bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning, storage of light pen 'hit' location, and address comparisons required for generation of timing signals and the split screen interrupt.

Buffer Control

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different 'handshaking' schemes are supported. These are described below.

SYSTEM CONFIGURATIONS

Figure 1 illustrates the block diagram of a typical display terminal using the Signetics 2670, 2671, 2672, and 2673 CRT terminal devices. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row.

The PVTC supports four common system configurations of display buffer memory, designated the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

The user programs bits 0 and 1 of IR0 to select the mode best suited for the system environment. The CNTRL1−3 outputs perform different functions for each mode and are named accordingly in the description of each mode.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by the signals read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non−contention type of operation that does not require address multiplexers. The CPU does not address the memory directly—the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The PVTC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

CRT TERMINAL BLOCK DIAGRAM

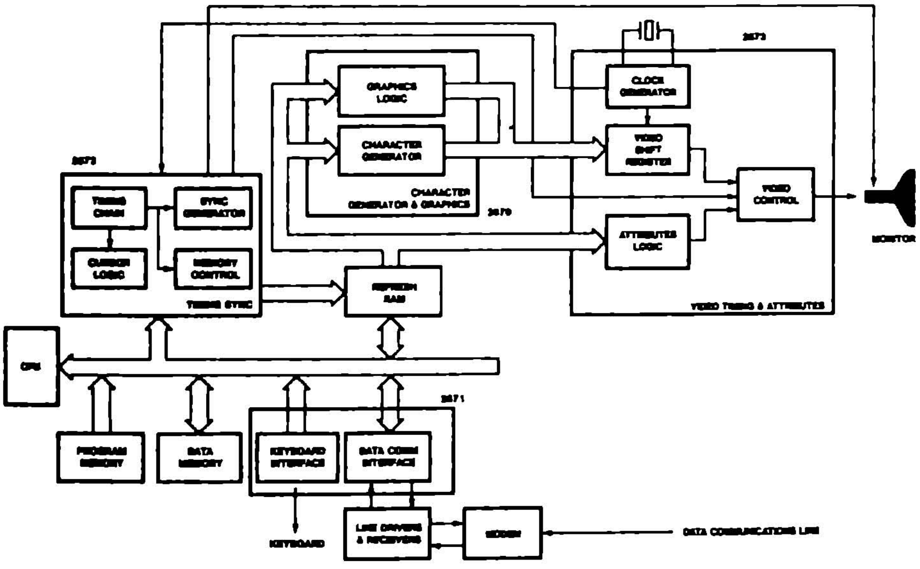

2673
TIMING CHAIN
SYNC GENERATOR
CURSOR LOGIC
MEMORY CONTROL
TIMING SYNC
GRAPHICS LOGIC
CHARACTER GENERATOR
CHARACTER GENERATOR & GRAPHICS
2670
2673
CLOCK GENERATOR
VIDEO SHIFT REGISTER
ATTRIBUTES LOGIC
VIDEO CONTROL
VIDEO TIMING & ATTRIBUTES
MONITOR
REFRESH RAM
CPU
2671
PROGRAM MEMORY
DATA MEMORY
KEYBOARD INTERFACE
DATA COMM INTERFACE
KEYBOARD
LINE DRIVERS & RECEIVERS
MODEM
DATA COMMUNICATIONS LINE

Figure 1

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SC2672

Preliminary

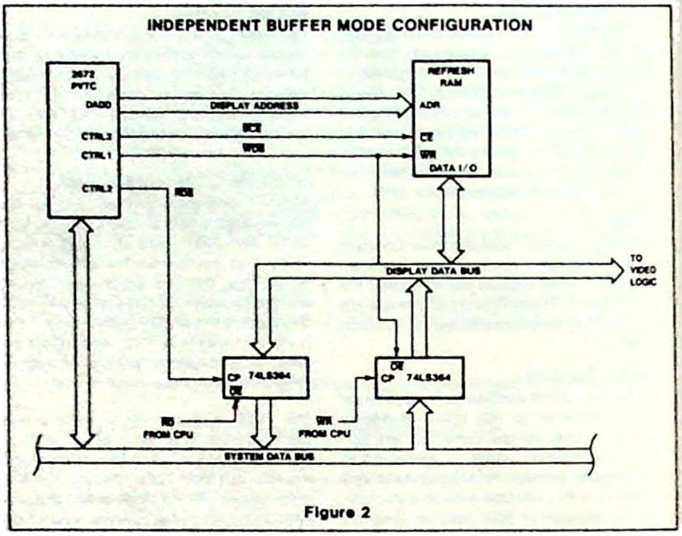


Figure 2

The CPU manages the data transfers by supplying commands to the PVTC. The commands used are:

1. Read/Write at pointer address.
2. Read/Write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.
4. CPU issues 'write at cursor with/without increment' or 'write at pointer' command.
5. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.
6. PVTC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1 and 3 as above.
2. CPU issues 'read at cursor with/without increment' or 'read at pointer' command.
3. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and PVTC sets RDFLG status to indicate that the read is completed.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

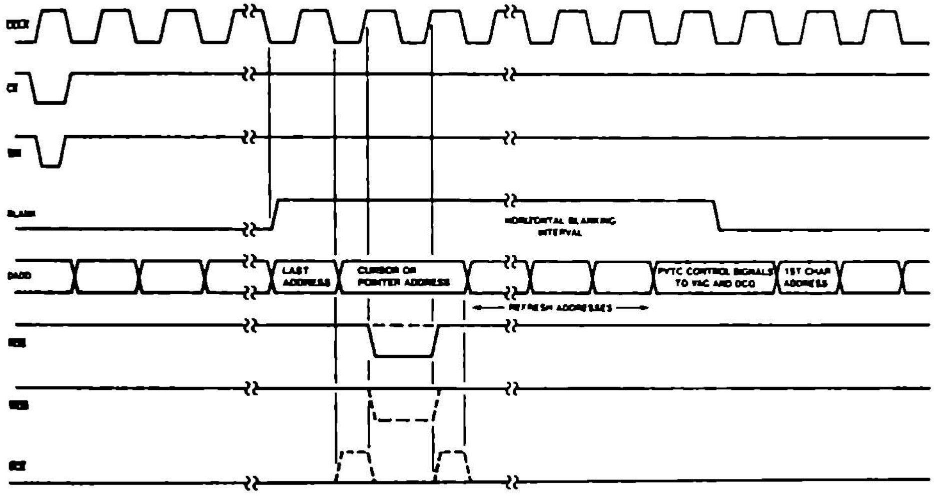
1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.
4. CPU issues 'write from cursor to pointer' command.
5. PVTC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. PVTC sets RDFLG status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously requested command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately one microsecond plus six (6) character clocks (see figure 4).

Timing for the 'write from cursor to pointer' operation is shown in figure 5. The BLANK output is asserted automatically and remains asserted until the vertical retrace interval following completion of the command. The memory is filled at a rate of one location per two character times, plus a small amount of overhead.

READ/WRITE AT CURSOR/POINTER COMMAND TIMING
(Command received during active display window)


BLANK
HORIZONTAL BLANKING INTERVAL
DADD
LAST ADDRESS
CURSOR OR POINTER ADDRESS
PVTC CONTROL SIGNALS TO VAC AND DCO
1ST CHAR ADDRESS
REFRESH ADDRESSES

NOTE
Write waveforms shown in dotted lines.

Figure 3

READ/WRITE AT CURSOR/POINTER COMMAND TIMING
(Command received while display is blanked)

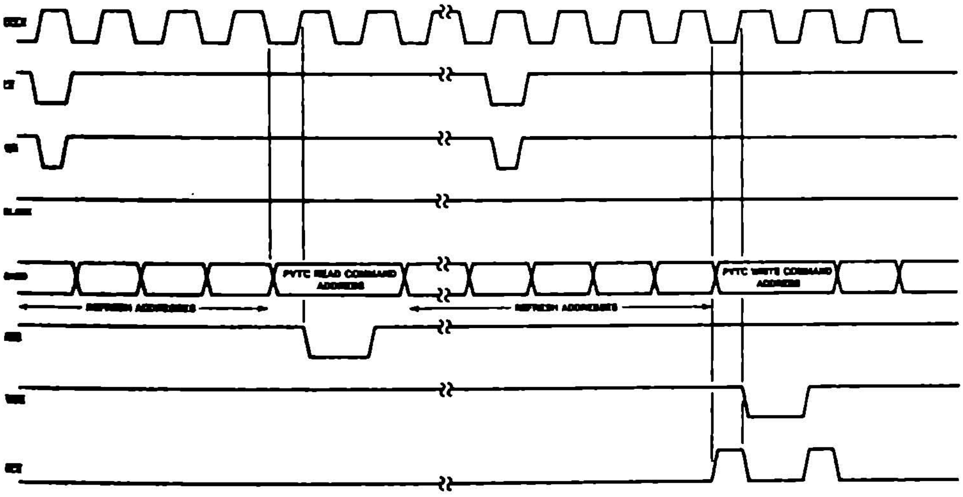

BLANK
DADD
PVTC READ COMMAND ADDRESS
PVTC WRITE COMMAND ADDRESS
REFRESH ADDRESSES
REFRESH ADDRESSES

Figure 4

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SC2672

Preliminary

WRITE FROM CURSOR TO POINTER COMMAND TIMING

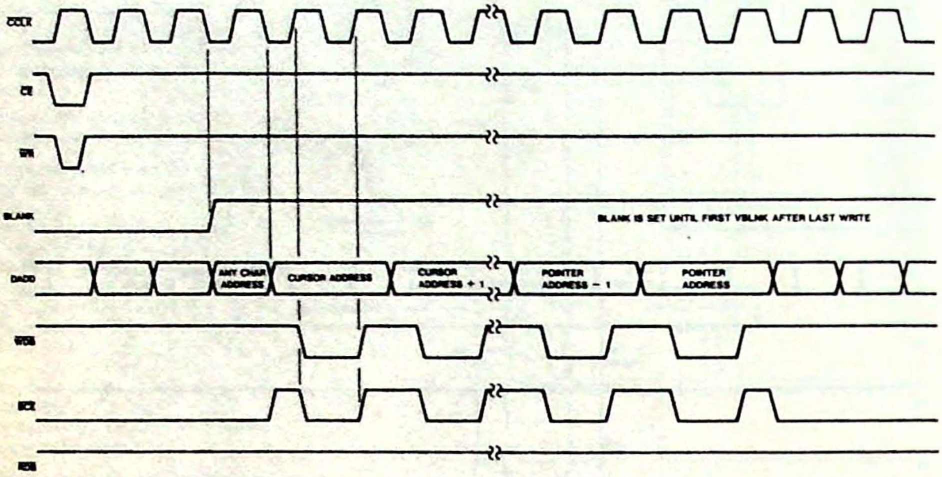

CCLK
CE
WR
BLANK
DADD
WDB
BCE
RDB
ANY CHAR ADDRESS
CURSOR ADDRESS
CURSOR ADDRESS + 1
POINTER ADDRESS − 1
POINTER ADDRESS
BLANK IS SET UNTIL FIRST VBLNK AFTER LAST WRITE

Figure 5

Shared and Transparent Buffer Modes

In these modes the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three-state drivers (see figure 6). The processor bus request ($\overline{\text{PBREQ}}$) control signal informs the PVTC that the CPU is requesting access to the display buffer. In response to this request, the PVTC raises bus acknowledge ($\overline{\text{BACK}}$) until its bus external ($\overline{\text{BEXT}}$) output has freed the display address and data busses for CPU access. $\overline{\text{BACK}}$, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the PVTC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the PVTC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in figures 7, 8, and 9.

PVTC SHARED OR TRANSPARENT BUFFER MODES

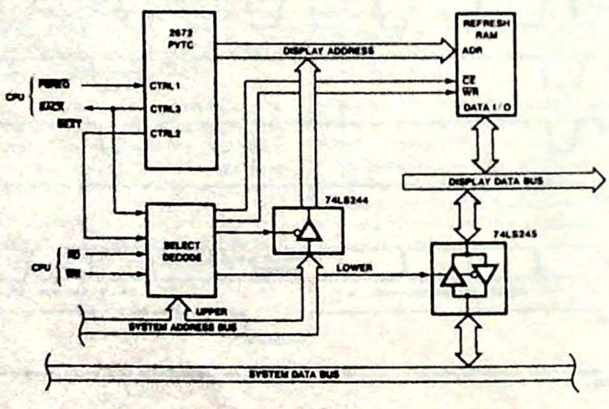
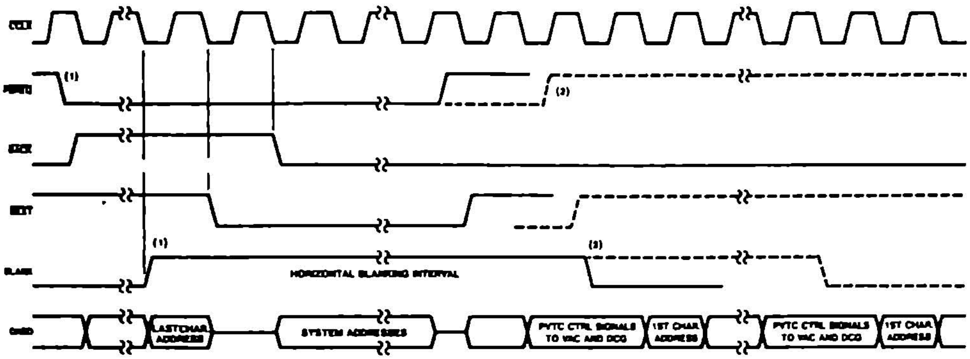

2672 PVTC
DISPLAY ADDRESS
REFRESH RAM
ADR
CE
WR
DATA I/O
CPU
PBREQ
BACK
BEXT
CTRL1
CTRL3
CTRL2
DISPLAY DATA BUS
74LS244
74LS245
SELECT DECODE
CPU
RD
WR
LOWER
UPPER
SYSTEM ADDRESS BUS
SYSTEM DATA BUS

Figure 6

TRANSPARENT BUFFER MODE TIMING

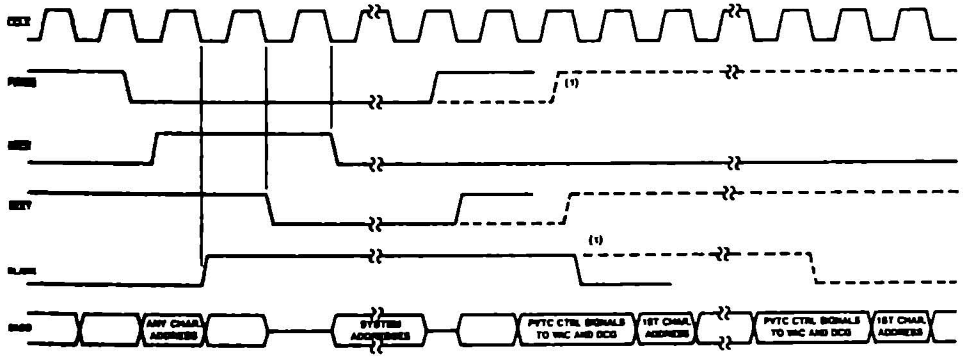

CCLK
PBREQ
BACK
BEXT
BLANK
DADD
(1)
(2)
(1)
(2)
HORIZONTAL BLANKING INTERVAL
LAST CHAR ADDRESS
SYSTEM ADDRESSES
PVTC CTRL SIGNALS TO VAC AND DCG
1ST CHAR ADDRESS
PVTC CTRL SIGNALS TO VAC AND DCG
1ST CHAR ADDRESS

NOTES

1. $\overline{\text{PBREQ}}$ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period
2. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 7

SHARED BUFFER MODE TIMING

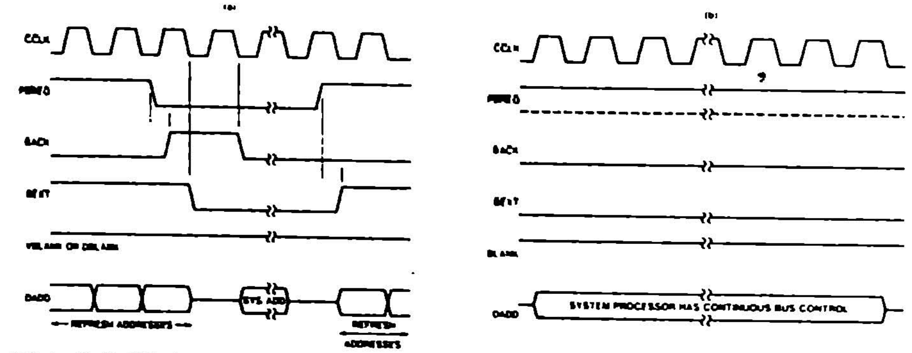

CCLK
PBREQ
BACK
BEXT
BLANK
DADD
(1)
(1)
ANY CHAR ADDRESS
SYSTEM ADDRESSES
PVTC CTRL SIGNALS TO VAC AND DCG
1ST CHAR ADDRESS
PVTC CTRL SIGNALS TO VAC AND DCG
1ST CHAR ADDRESS

NOTE

1. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked

Figure 8

SHARED AND TRANSPARENT MODE TIMING


(a)
CCLK
MREQ
BACK
BEXT
VBLANK OR DBLANK
DADD
SYS ADD
REFRESH ADDRESSES
REFRESH ADDRESSES
(b)
CCLK
MREQ
BACK
BEXT
BLANK
DADD
SYSTEM PROCESSOR HAS CONTINUOUS BUS CONTROL

a) During Vertical Blank or after 'display off' command

b) After 'display off and 3-state' command

Figure 9

Row Buffer Mode

Figures 10 and 11 show the timing and a typical hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the PVTC halts the CPU and DMA's the next row of character data from the system memory to the row buffer memory. The PVTC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The bus request control ($\overline{\text{BREQ}}$) signal informs the CPU that character addresses and the memory bus control (MBC) signal will start at the next falling edge of BLANK. The CPU must release the address and data busses before this time to prevent bus contention. After the row of character data is transferred to the CPU, $\overline{\text{BREQ}}$ returns high to grant memory control back to the CPU.

ROW BUFFER MODE CONFIGURATION

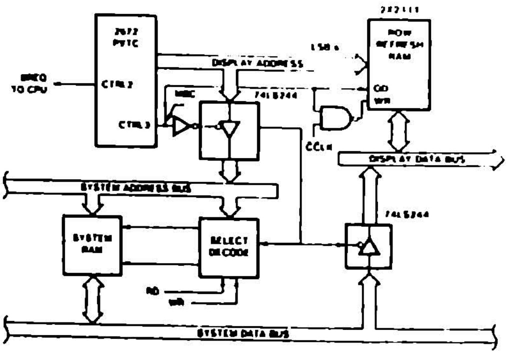

2672 PVTC
BREQ TO CPU
CTRL 2
CTRL 3
MBC
DISPLAY ADDRESS
74LS244
LSB's
2X2111
ROW REFRESH RAM
OD
WR
CCLK
DISPLAY DATA BUS
SYSTEM ADDRESS BUS
SYSTEM RAM
SELECT DECODE
74LS244
RD
WR
SYSTEM DATA BUS

Figure 10

ROW BUFFER MODE TIMING

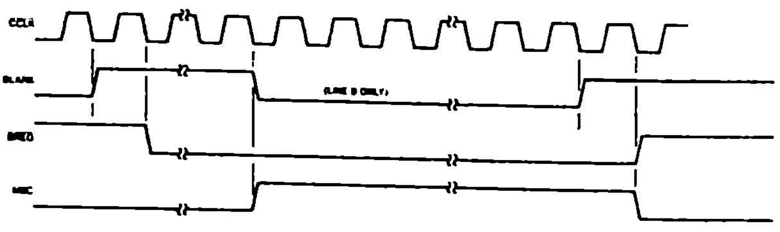

CCLK
BLANK
(LINE 0 ONLY)
BREQ
MBC

Figure 11

OPERATION

After power is applied, the PVTC will be in an inactive state. Two consecutive 'master reset' commands are necessary to release this circuitry and ready the PVTC for operation. Two register groups exist within the PVTC: the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, and screen format. These are loaded first and normally require no modification except for certain special visual effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the pointer address for independent memory access mode. These usually require modification during operation.

After initial loading of the two register groups, the PVTC is ready to control the monitor screen. Prior to executing the PVTC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the PVTC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the PVTC display control and command registers, and the initialization registers, if required. Interrupts and status conditions generated by the PVTC supply the 'handshaking' information necessary for the CPU to effect the display changes in the proper time frame.

INITIALIZATION REGISTERS

There are 11 initialization registers (IR0–IR10) which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split screen register) is accessed. The pointer then continues to point to the split screen register. Upon power-up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in figure 12.

INITIALIZATION REGISTER FORMATS

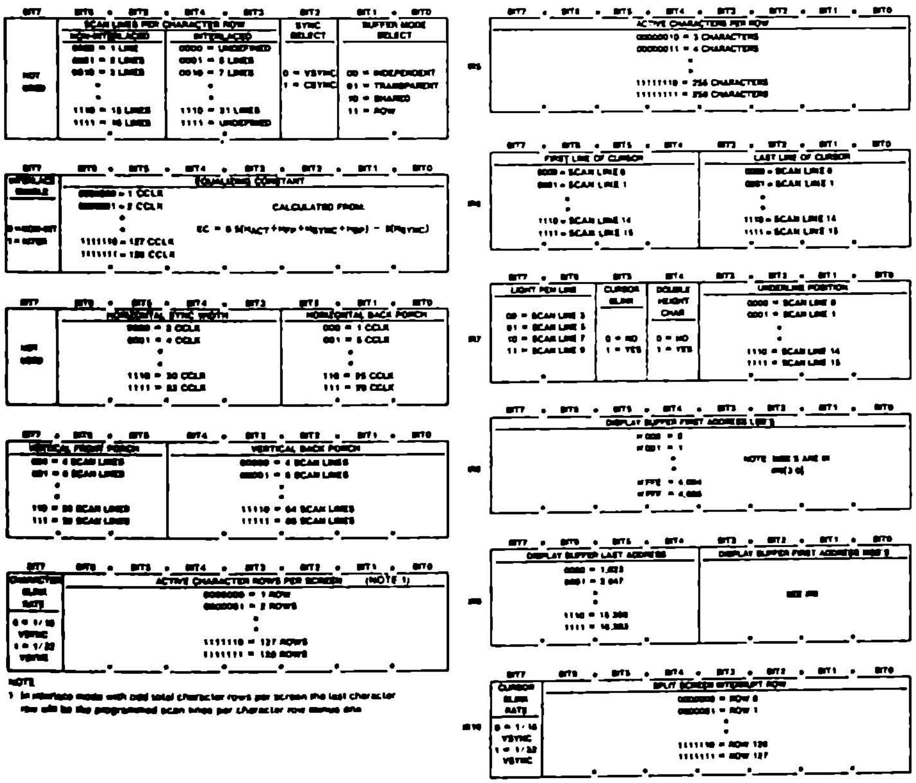

BIT7 BIT6 BIT5 BIT4 BIT3 BIT2 BIT1 BIT0
IR0
NOT USED
SCAN LINES PER CHARACTER ROW
NON-INTERLACED
INTERLACED
1110 = 15 LINES
1111 = 16 LINES
1110 = 31 LINES
1111 = UNDEFINED
SYNC SELECT
0 = VSYNC
1 = CSYNC
BUFFER MODE SELECT
00 = INDEPENDENT
01 = TRANSPARENT
10 = SHARED
11 = ROW
IR1
EQUALIZING CONSTANT
CALCULATED FROM
1111110 = 127 CCLK
1111111 = 128 CCLK
IR2
HORIZONTAL SYNC WIDTH
HORIZONTAL BACK PORCH
000 = 1 CCLK
IR3
VERTICAL FRONT PORCH
VERTICAL BACK PORCH
IR4
CHARACTER BLINK RATE
ACTIVE CHARACTER ROWS PER SCREEN (NOTE 1)
0000000 = 1 ROW
1111110 = 127 ROWS
1111111 = 128 ROWS
NOTE
1 In interlace mode with odd total character rows per screen the last character row will be the programmed scan lines per character row minus one
IR5
ACTIVE CHARACTERS PER ROW
IR6
FIRST LINE OF CURSOR
LAST LINE OF CURSOR
IR7
LIGHT PEN LINE
CURSOR BLINK
DOUBLE HEIGHT CHAR
UNDERLINE POSITION
0 = NO
1 = YES
IR8
DISPLAY BUFFER FIRST ADDRESS LSB'S
IR9
DISPLAY BUFFER LAST ADDRESS
DISPLAY BUFFER FIRST ADDRESS MSB'S
IR10
CURSOR BLINK RATE
SPLIT SCREEN INTERRUPT ROW
0 = 1/16 VSYNC
1 = 1/32 VSYNC

Figure 12

IR0[6:3]—Scan Lines per Character Row

Both interlaced and non-interlaced scanning are supported by the PVTC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the PVTC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0–LA3 and LI pins.

IR0[2]—VS/CS Enable

This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

IR0[1:0]—Buffer Mode Select

Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. See System Configuration.

IR1[7]—Interlace Enable

Specifies interlaced or noninterlaced timing operation. Two modes of interlaced operation are available, depending on whether L0–L3 or LI, L0–L2 are used as the line address for the character generator. The resulting displays are shown in figure 13.

For 'interlaced sync' operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The PVTC outputs successive line numbers in ascending order on the LA0–LA3 lines, one per scan line for each field.

The 'interlaced sync and video' format doubles the character density on the screen. The PVTC outputs successive line numbers in ascending order on the LI, LA0–LA2 lines, one per scan line for each field, but alternates beginning the count with even and odd line numbers. This displays the odd field with even scan lines in even character rows and odd scan lines in odd character rows, and the even field with odd scan lines in even character rows and even scan lines on odd character rows. This provides balanced beam currents in the odd and even fields, thus minimizing character variations due to different loading of the CRT anode supply between fields.

IR1[6:0]—Equalizing Constant

This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{CCLK}$) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}}{2} - 2(H_{SYNC})$$

The definition of the individual parameters is illustrated in figure 14.

Note that when using the 2673 VAC, it will delay the blank pulse three $\overline{CCLKs}$ relative to the HSYNC pulse.

IR2[6:3]—Horizontal Sync Pulse Width

This field specifies the width of the HSYNC pulse in $\overline{CCLK}$ periods.

IR2[2:0]—Horizontal Back Porch

This field defines the number of CCLKs between the trailing edge of HSYNC and the trailing edge of BLANK.

IR3[7:5]—Vertical Front Porch

Programs the number of scan line periods between the rising edges of BLANK and VSYNC during a vertical retrace interval. The width of the VSYNC pulse is fixed at three scan lines.

IR3[4:0]—Vertical Back Porch

This field determines the number of scan line periods between the falling edges of the VSYNC and BLANK outputs.

IR4[7]—Character Blink Rate

Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/16 or 1/32 of the vertical field rate. The timing signal has a duty cycle of 75% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

IR4[6:0]—Character Rows Per Screen

This field defines the number of character rows to be displayed. This value multiplied by the scan lines per character row, plus the vertical front and back porch values, and the vertical sync pulse width (three scan lines) is the vertical scan period in scan lines.

IR5[7:0]—Active Characters Per Row

This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period in $\overline{CCLKs}$.

IR6[7:4], IR6[3:0]—First and Last Scan Line of Cursor

These two fields specify the height and position of the cursor on the character block. The 'first' line is the topmost line when scanning from the top to the bottom of the screen.

IR7[7:6]—Light Pen Line Position

This field defines which of four scan lines of the character row will be used for the light pen strike-thru attribute by the 2673 VAC. The timing signal is multiplexed onto the DADD9/LPL output during the falling edge of BLANK.

IR7[5]—Cursor Blink Enable

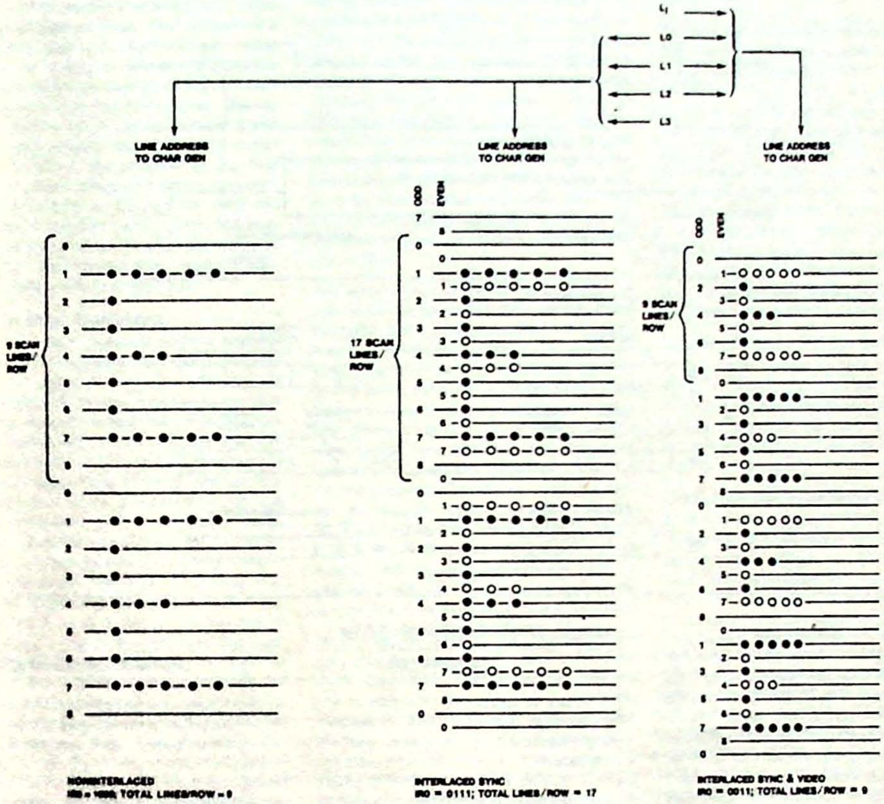
This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR10[7]). The blink duty cycle for the cursor is 50%.

IR7[4]—Double Height Character Row Enable

If enabled, the number of each scan line will be repeated twice in succession, causing the height of the character row to double. This bit can be changed at any time but will only become effective at the beginning of the character row following the time it is changed. This allows selected character rows to be of double height. The split screen interrupt can be used to notify the CPU when to effectuate changes to this bit. For each double height row which replaces a normal row, one row count should be subtracted from the 'character rows per screen' field (IR4) to maintain the same total number of scan lines per field.

IR7[3:0]—Underline Position

This field defines which scan line of the character row will be used for the underline attribute by the 2673 VAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.



NONINTERLACED
IR0 = 1000; TOTAL LINES/ROW = 9

INTERLACED SYNC
IR0 = 0111; TOTAL LINES/ROW = 17

INTERLACED SYNC & VIDEO
IR0 = 0011; TOTAL LINES/ROW = 9

Figure 13

IR9[3:0], IR8[7:0]—Display Buffer First Address

IR9[7:4]—Display Buffer Last Address

These two fields define the area within the buffer memory where the display data will reside. When the data at the 'display buffer last address' is displayed, the PVTC will wrap-around and obtain the data to be displayed at the next screen position from the 'display buffer first address'. If 'last address' is the end of a character row and a new screen start address has been loaded into the screen start register, or if 'last address' is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

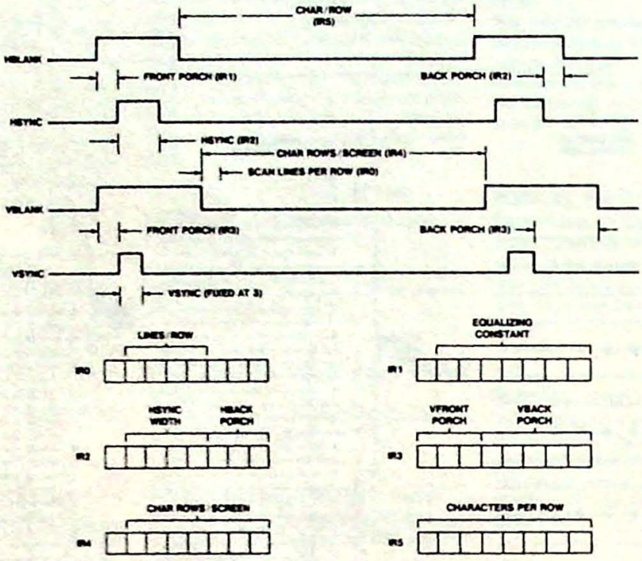
Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the split screen interrupt feature of the PVTC.

IR10[7] – Cursor Blink Rate

The cursor blink rate can be specified at 1/16 or 1/32 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

IR10[6:0]—Split Screen Interrupt

The split screen interrupt can be used to provide special screen effects such as a row of double height characters or to change the normal addressing sequence of the display memory. The contents of this field is compared, in real time, to the current character row number. Upon a match, the PVTC sets the split screen status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of scan line zero of the split screen character row.


HORIZONTAL AND VERTICAL TIMING

CHAR/ROW
(IR5)
HBLANK
FRONT PORCH (IR1)
BACK PORCH (IR2)
HSYNC
HSYNC (IR2)
CHAR ROWS/SCREEN (IR4)
SCAN LINES PER ROW (IR0)
VBLANK
FRONT PORCH (IR3)
BACK PORCH (IR3)
VSYNC
VSYNC (FIXED AT 3)
LINES/ROW
IR0
EQUALIZING CONSTANT
IR1
HSYNC WIDTH
HBACK PORCH
IR2
VFRONT PORCH
VBACK PORCH
IR3
CHAR ROWS/SCREEN
IR4
CHARACTERS PER ROW
IR5

Figure 14

Timing Considerations

Normally, the contents of the initialization registers are not changed during operation. However, this may be necessary to implement special display features such as multiple cursors, smooth scrolling, horizontal scrolling, and double height character rows. Table 2 describes timing details for these registers which should be considered when implementing these features.

Table 2 TIMING CONSIDERATIONS

PARAMETER	TIMING CONSIDERATIONS
First line of cursor Last line of cursor Light pen line Underline	These parameters must be established at a minimum of two character times prior to their occurence.
Double height characters	Set/reset during the character row prior to the row which is to be/not to be double height
Cursor blink Cursor blink rate Character blink rate	New values become effective within one field after values are changed
Split screen interrupt row	Change anytime prior to line zero of desired row
Character rows per screen	Change only during vertical blanking period
Vertical front porch	Change prior to first line of V_{FP}
Vertical back porch	Change prior to fourth line after V_{SYNC}
Screen start register	Change prior to the horizontal blanking interval of the last line of character row before row where new value is to be used

DISPLAY CONTROL REGISTERS

There are nine registers in this group, each with an individual address. Their formats are illustrated in figure 15. The command register is used to invoke one of 16 possible PVTC commands as described in the COMMANDS section of this data sheet. The remaining registers in the group store address values which specify the cursor and buffer pointer locations, the location of the first character to be displayed on the screen, and the location of a light pen 'hit'. With the exception of the light pen register, the user initializes these registers after powering on the system and changes their values to control the data which is displayed.

Screen Start Registers

The screen start registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row this address is transferred to the row start register (RSR) and into the memory address counter (MAC). The counter is then incremented sequentially at the character rate the number of times programmed into the active characters per row register (IR5), thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC is loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval, the entire process repeats again.

The sequential operation described above will be modified upon the occurence of either of two events. First, if during the incrementing of the memory address counter the 'display buffer last address' (IR9[7:4]) is reached, the MAC will be loaded from the 'display buffer first address' register (IR9[3:0], IR8[7:0]) at the next character clock. Sequential operation will then resume starting from this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see figure 16a).

The sequential row to row addressing can also be modified under CPU control. If the contents of the screen start register (upper, lower, or both) are changed during any character row (say row 'n'), the starting address of the next character row (row 'n + 1') will be the new value of the screen start register and addressing will continue sequentially from there. This allows features such as split screen operation, partial scroll, or status line display to be implemented. The split screen interrupt feature of the PVTC is useful in controlling this type of operation. Note that in order to obtain the correct screen display, the screen start register must be reloaded with the original value prior to the end of the vertical retrace. See figure 16b.

During vertical blanking the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free-run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered, refreshing continues from the display buffer first address.

Cursor Address Registers

The contents of these registers defines the buffer memory address of the cursor. If enabled, the cursor output will be asserted when the memory address counter matches the value of the cursor address registers. The cursor address registers may be read or written by the CPU or incremented via the 'increment cursor address' command. In independent buffer mode, these registers define a buffer memory address for PVTC controlled access in response to 'read/write at cursor with/without increment' commands, or the first address to be used in executing the 'write from cursor to pointer' command.

Display Pointer Address Registers

These registers define a buffer memory address for PVTC controlled accesses in response to 'read/write at pointer' commands. They also define the last buffer memory address to be written for the 'write from cursor to pointer' command.

Light Pen Address Registers

If the light pen input is enabled, these registers are used to store the current character address upon receipt of a light pen strobe input. Several sources of delay between the display of a character upon the screen and the receipt of a light pen hit can be expected to exist in a system environment. These delays include address pipelining in the character generation circuits, delays in the video generation circuits, and delays in the light detection circuitry itself. These delays cause the value stored in the light pen register to differ from the actual address of the character at which the light pen hit actually was detected. Software must be used to correct this condition.

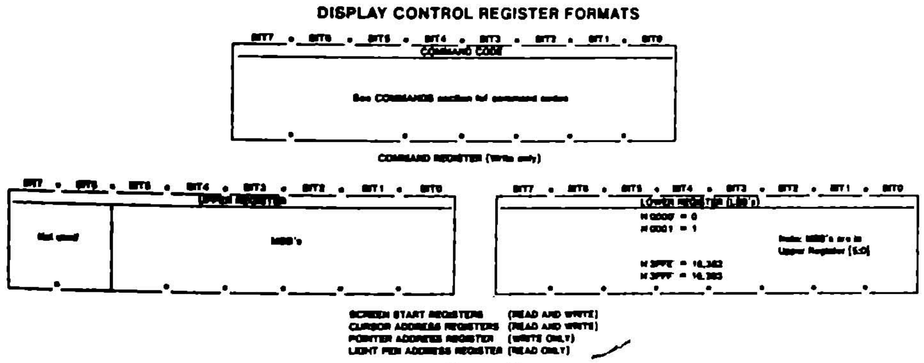

DISPLAY CONTROL REGISTER FORMATS
BIT7
BIT6
BIT5
BIT4
BIT3
BIT2
BIT1
BIT0
COMMAND CODE
See COMMANDS section for command codes
COMMAND REGISTER (Write only)
UPPER REGISTER
Not used
MSB's
LOWER REGISTER (LSB's)
H 0000 = 0
H 0001 = 1
H 3FFE = 16,382
H 3FFF = 16,383
Note: MSB's are in Upper Register [5:0]
SCREEN START REGISTERS (READ AND WRITE)
CURSOR ADDRESS REGISTERS (READ AND WRITE)
POINTER ADDRESS REGISTER (WRITE ONLY)
LIGHT PEN ADDRESS REGISTER (READ ONLY)

Figure 15

Preliminary

INTERRUPT/STATUS REGISTERS

The interrupt and status registers provide information to the CPU to allow it to interact with the PVTC to effect desired changes to implement various display operations. The interrupt register provides information on five possible interrupting conditions, as shown in figure 17. These conditions may be selectively enabled or disabled (masked) from causing interrupts by certain PVTC commands. An interrupt condition which is enabled (mask bit equal to one) will cause the $\overline{\text{INTR}}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set upon occurrence of the interrupting condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{\text{INTR}}$ output or the interrupt register.

The status register provides six bits of status information: the five possible interrupting conditions plus the NOT BUSY bit. For this register, however, the contents are not effected by the state of the mask bits.

Descriptions of each interrupt/status register bit follow. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a 'reset interrupt/status bits' command. The bits are also reset by a 'master reset' command and upon power-up

SR[5] – RDFLG

This bit is present in the status register only. A zero indicates that the PVTC is currently executing the previously issued command. A one indicates that the PVTC is ready to accept a new command.

I/SR[4] – VBLANK

Indicates the beginning of a vertical blanking interval. Is set to a one at the beginning of the first scan line of the vertical front porch.

I/SR[3] – Line Zero

Is set to a one at the beginning of the first scan line (line 0) of each active character row.

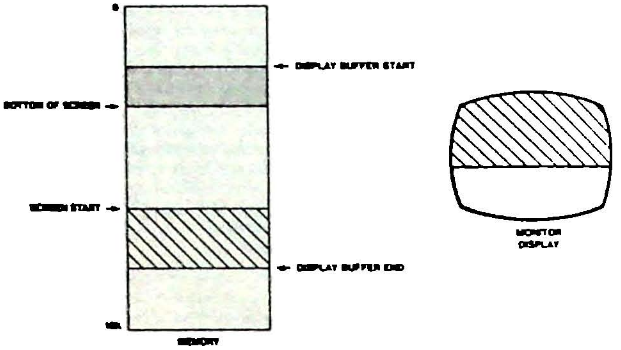
I/SR[2] – Split Screen

This bit is set when a match occurs between the current character row number and the value contained in the split screen interrupt register, IR10[6:0]. The equality condition is only checked at the beginning of line zero of each character row. This bit is reset when either of the screen start registers is loaded by the CPU.

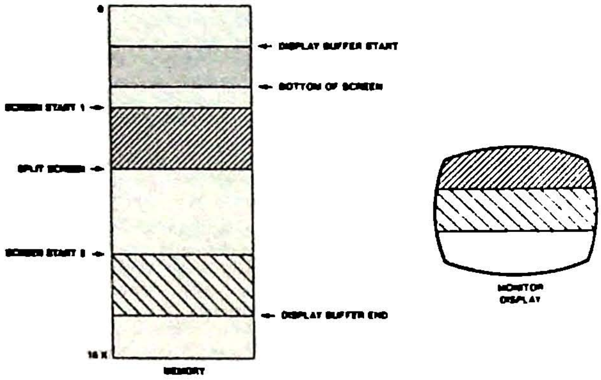
I/SR[1] – Ready

Certain PVTC commands affect the display and may require the PVTC to wait for a blanking interval before enacting the command. This bit is set to one when execution of the command has been completed. No command should be invoked until the prior command is completed.

DISPLAY ADDRESSING OPERATION



(a) DISPLAY MEMORY WRAPAROUND



(b) DISPLAY MEMORY SPLIT SCREEN WITH WRAPAROUND

Figure 16

INTERRUPT AND STATUS REGISTER FORMAT

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Not used always read as 0		RDFLG	VBLANK	LINE ZERO	SPLIT SCREEN	READY	LIGHT PEN
		0 = Busy 1 = Ready	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Busy 1 = Ready	0 = No 1 = Yes

NOTE

*Status register only. Always 0 when reading interrupt register

Figure 17

I/SR[0] – Light Pen

A one indicates that a light pen hit has occurred and that the contents of the light pen register have been updated. This bit will be reset when either of the light pen registers is read.

COMMANDS

The PVTC commands are divided into two classes: the instantaneous commands, which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. Command formats are shown in table 3. The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

Instantaneous Commands

The instantaneous commands are executed immediately after the trailing edge of the $\overline{WR}$ pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits. However, a command should not be invoked till the RDFLG bit is low.

Master Reset

This command initializes the PVTC and may be invoked at any time to return the PVTC to its initial state. Upon power-up, two successive master reset commands must be applied to release the PVTC's internal power on circuits. In transparent and shared buffer modes, the CNTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of RESET and BLANK goes high. BLANK remains high until a 'display on' command is received.
2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The transparent mode, cursoroff, display off, and light pen disable states are set.
4. The initialization register pointer is set to address IR0.

Load IR Address

This command is used to preset the initialization register pointer with the value 'V' defined by D3–D0. Allowable values are 0 to 10.

Enable Light Pen

After invoking this command, receipt of a light pen strobe input will cause the light pen register to be loaded with the current buffer memory address and the corresponding interrupt and status flag to be set. Once loaded, further loads are inhibited until either one of the light pen registers are read or a reset function is performed.

Disable Light Pen

Light pen hits will not be recognized.

Display Off

Asserts the BLANK output. The DADD0 thru DADD13 display address bus outputs may be optionally placed in the three–state condition by setting bit 2 to a '1' when invoking the command.

Table 3. PVTC Command Formats

D7	D6	D5	D4	D3	D2	D1	D0	Hex	COMMAND
Instantaneous Commands:									
0	0	0	0	0	0	0	0		Master reset
0	0	0	1	V	V	V	V		Load IR pointer with value V (V = 0 to 10)
0	0	1	d	d	d	1	0[1]		Disable light pen
0	0	1	d	d	d	1	1[2]		Enable light pen
0	0	1	d	1	N	d	0[1]		Display off. Float DADD bus if N = 1
0	0	1	d	1	N	d	1[2]		Display on: Next field (N = 1) or scan line (N = 0)
0	0	1	1	d	d	d	0[1]		Cursor off
0	0	1	1	d	d	d	1[2]		Cursor on
0	1	0	N	N	N	N	N		Reset interrupt/status: Bit reset where N = 1
1	0	0	N	N	N	N	N		Disable interrupt: Disable where N = 1
0	1	1	N	N	N	N	N		Enable interrupt: Enables interupts and resets the corresponding interrupt/status bits where N = 1
			V B	L Z	S S	R D	L P		
Delayed Commands:								**Hex**	
1	0	1	0	0	1	0	0	A4	Read at pointer address
1	0	1	0	0	0	1	0	A2	Write at pointer address
1	0	1	0	1	0	0	1	A9	Increment cursor address
1	0	1	0	1	1	0	0	AC	Read at cursor address
1	0	1	0	1	0	1	0	AA	Write at cursor address
1	0	1	0	1	1	0	1	AD	Read at cursor address and increment address
1	0	1	0	1	0	1	1	AB	Write at cursor address and Increment address
1	0	1	1	1	0	1	1	BB	Write from cursor address to pointer address

NOTES
1 Any combination of these three commands is valid
2 Any combination of these three commands is valid
3 d = don't care

Display On

Restores normal blanking operation either at the beginning of the next field (bit 2 = 1) or at the beginning of the next scan line (bit 2 = 0). Also returns the DADD0–DADD13 drivers to their active state.

Cursor Off

Disables cursor operation. Cursor output is placed in the low state.

Cursor On

Enables normal cursor operation.

Reset Interrupt/Status Bits

This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 – Light pen
Bit 1 – Ready
Bit 2 – Split screen
Bit 3 – Line zero
Bit 4 – Vertical blank

Disable Interrupts

Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from asserting the $\overline{INTR}$ output. Bit position correspondence is as above.

Enable Interrupts

Resets the selected interrupt and status register bits and writes the associated interrupt mask bits to a one. This enables the corresponding conditions to assert the $\overline{INTR}$ output. Bit position correspondence is as above.

Delayed Commands

This group of commands is utilized for the independent buffer mode of operation, although the 'increment cursor' command can also be used in other modes. With the exception of the 'write from cursor to pointer' and 'increment cursor' commands, all the commands of this type will be executed immediately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking interval. If invoked during a vertical retrace interval or a 'display off' state, the command is executed immediately.

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SC2672

Preliminary

The 'increment cursor' and 'write from cursor to pointer' commands are executed immediately after they are issued. 'Increment cursor' requires approximately three $\overline{\text{CCLK}}$ periods for completion. 'Write from cursor to pointer' asserts the BLANK output during its execution. BLANK will not be released until the beginning of the vertical blanking interval following the last write operation. A second 'write from cursor to pointer' command should not be issued until this time.

In all cases, the PVTC will assert the READY/RDFLG status to signify completion of the command. No other commands should be given until the current command is completed. Therefore, the READY interrupt or RDFLG status flag should be used for handshaking control between the PVTC and CPU when using these commands.

Read/Write at Pointer

Transfers data between the display buffer and the bus interface latch using the address contained in the pointer register.

Read/Write at Cursor

Transfers data between the display buffer and the bus interface latch using the address contained in the cursor register.

Increment Cursor

Adds one (modulo 16K) to the cursor address register.

Read/Write at Cursor and Increment

Transfers data between the display buffer and the bus interface latch using the address contained in the cursor register and then adds one (modulo 16K) to the cursor address register.

Write from Cursor to Pointer

Writes the data contained in the bus interface latch into the block of display memory designated by the the cursor address and pointer address registers, inclusive. After completion of the command, the pointer address will be unchanged, but the cursor register contents will be equal to the pointer address.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage	$I_{OL} = 1.6mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{\text{INTR}}$ output)	$I_{OH} = -100 uA$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	µA
I_{LL}	Data bus 3−state leakage current	$V_O = 0$ to V_{CC}	−10		10	µA
I_{OD}	$\overline{\text{INTR}}$ open drain output leakage current	$V_O = 0$ to V_{CC}			10	µA
I_{CC}	Power supply current				160	mA

NOTES
See next page

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SC2672

Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6,7,8]

PARAMETER		TEST CONDITIONS	TENTATIVE LIMITS			UNIT
			Min	Typ	Max	
Bus Timing (Fig. 18)[9]						
t_{AS}	A0–A2 setup time to $\overline{WR}$, $\overline{RD}$ low		30			ns
t_{AH}	A0–A2 hold time from $\overline{WR}$, $\overline{RD}$ high		0			ns
t_{CS}	$\overline{CE}$ setup time to $\overline{WR}$, $\overline{RD}$ low		0			ns
t_{CH}	$\overline{CE}$ hold time from $\overline{WR}$, $\overline{RD}$ high		0			ns
t_{RW}	$\overline{WR}$, $\overline{RD}$ pulse width		250			ns
t_{DD}	Data valid after $\overline{RD}$ low				200	ns
t_{DF}	Data bus floating after $\overline{RD}$ high				100	ns
t_{DS}	Data setup time to $\overline{WR}$ high		150			ns
t_{DH}	Data hold time from $\overline{WR}$ high		0			ns
t_{CC}	Time between READs and/or WRITEs[10]		600			ns
$\overline{CCLK}$ Timing (Fig. 19)						
t_{CCP}	$\overline{CCLK}$ period		250			ns
t_{CCH}	$\overline{CCLK}$ high time		100			ns
t_{CCL}	$\overline{CCLK}$ low time		100			ns
t_{CCD}	Output delay from $\overline{CCLK}$ edge				150	ns
Other Timings (Fig. 20)						
t_{RDL}	READY/RDFLG low from $\overline{WR}$ high[9]				600	ns
t_{BAK}	$\overline{BACK}$ high from $\overline{PBREQ}$ low				150	ns
t_{BXT}	$\overline{BEXT}$ high from $\overline{PBREQ}$ high				150	ns
t_{LPS}	Light pen strobe setup time to $\overline{CCLK}$ low		120			ns
t_{LPH}	Light pen strobe hold time from $\overline{CCLK}$ low		−10			ns
t_{IRL}	$\overline{INTR}$ low from $\overline{CCLK}$ low				150	ns
t_{IRH}	$\overline{INTR}$ high from $\overline{WR}$, $\overline{RD}$ high[9]				600	ns

Notes

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 65°C/W junction to ambient ([illegible] ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND).
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
8. Test condition for outputs: $C_L = 150pF$.
9. Timing is illustrated and specified referenced to $\overline{WR}$ and $\overline{RD}$ inputs. Device may also be operated with $\overline{CE}$ as the 'strobing' input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{CE}$.
10. 1 microsecond minimum after 'master reset' command.

BUS TIMING

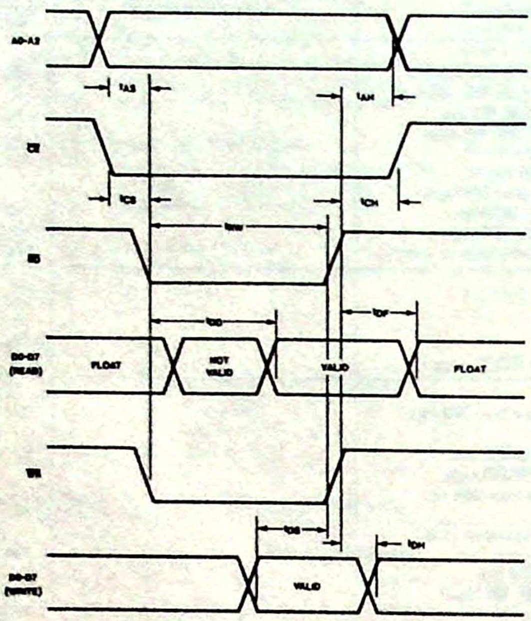
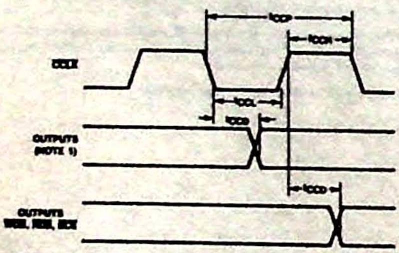

A0-A2
tAS
tAH
CE
tCS
tCH
RD
tRW
tDD
tDF
D0-D7
(READ)
FLOAT
NOT VALID
VALID
FLOAT
WR
tDS
tDH
D0-D7
(WRITE)
VALID

Figure 18

$\overline{CCLK}$ TIMING


tCCP
tCCH
CCLK
tCCL
tCCD
OUTPUTS
(NOTE 1)
tCCD
OUTPUTS

NOTES

1 DADD0-DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BCE}$, MBC, BACK

2 $\overline{BCE}$ changes state on both $\overline{CCLK}$ edges—(see Figures 3 and 4).

Figure 19

OTHER TIMINGS

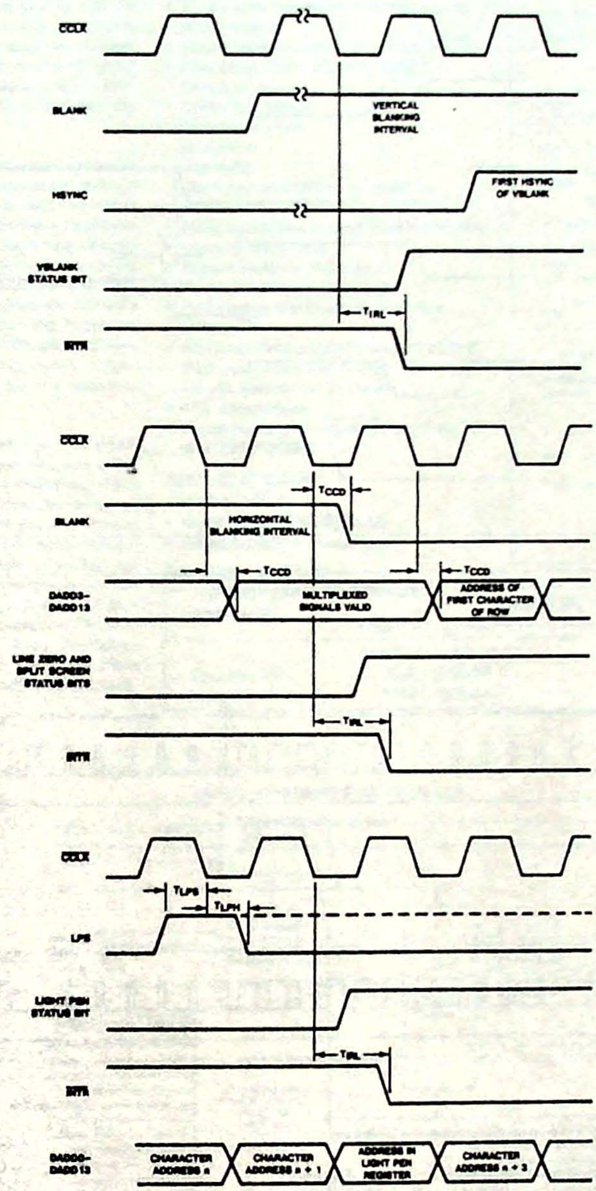

CCLK
BLANK
VERTICAL BLANKING INTERVAL
HSYNC
FIRST HSYNC OF VBLANK
VBLANK STATUS BIT
T_IRL
INTR
CCLK
T_CCD
BLANK
HORIZONTAL BLANKING INTERVAL
T_CCD
T_CCD
DADD3-DADD13
MULTIPLEXED SIGNALS VALID
ADDRESS OF FIRST CHARACTER OF ROW
LINE ZERO AND SPLIT SCREEN STATUS BITS
T_IRL
INTR
CCLK
T_LPS
T_LPH
LPS
LIGHT PEN STATUS BIT
T_IRL
INTR
DADD0-DADD13
CHARACTER ADDRESS n
CHARACTER ADDRESS n + 1
ADDRESS IN LIGHT PEN REGISTER
CHARACTER ADDRESS n + 3

Figure 20

OTHER TIMINGS (Continued)

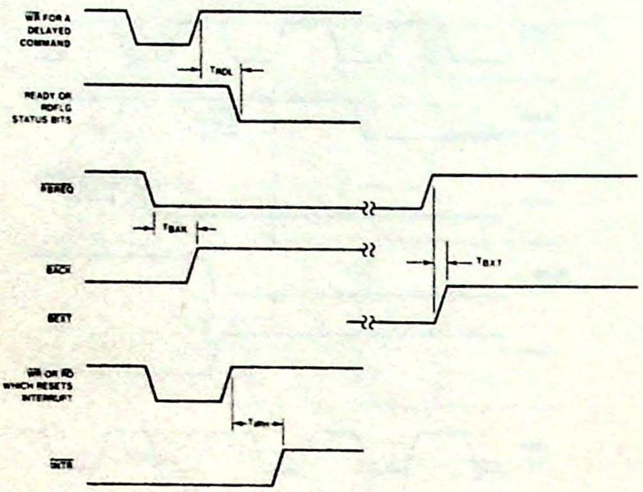
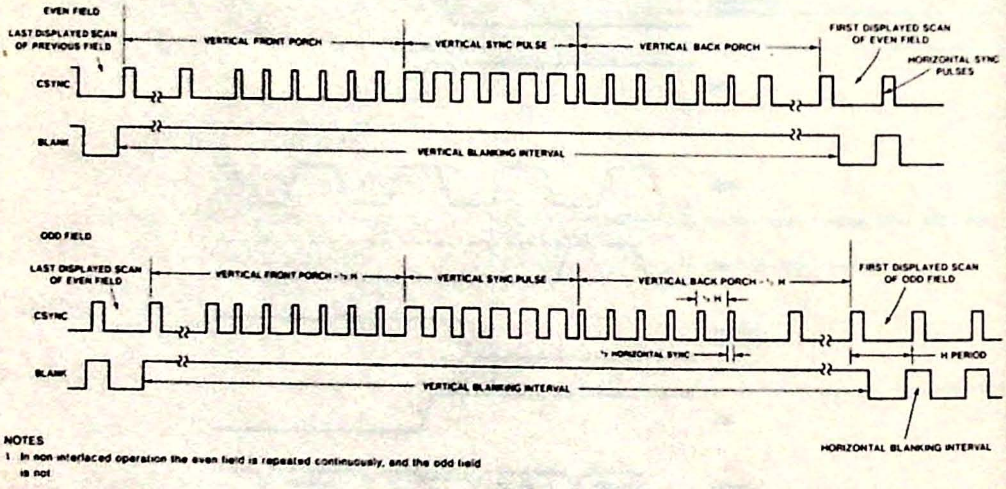

WR FOR A DELAYED COMMAND
T_RDL
READY OR RDFLG STATUS BITS
PBREQ
T_BAK
BACK
T_BXT
BEXT
WR OR RD WHICH RESETS INTERRUPT
T
INTR

Figure 20

COMPOSITE SYNC TIMING


EVEN FIELD
LAST DISPLAYED SCAN OF PREVIOUS FIELD
VERTICAL FRONT PORCH
VERTICAL SYNC PULSE
VERTICAL BACK PORCH
FIRST DISPLAYED SCAN OF EVEN FIELD
HORIZONTAL SYNC PULSES
CSYNC
BLANK
VERTICAL BLANKING INTERVAL
ODD FIELD
LAST DISPLAYED SCAN OF EVEN FIELD
VERTICAL FRONT PORCH + ½ H
VERTICAL SYNC PULSE
VERTICAL BACK PORCH + ½ H
½ H
FIRST DISPLAYED SCAN OF ODD FIELD
CSYNC
½ HORIZONTAL SYNC
H PERIOD
BLANK
VERTICAL BLANKING INTERVAL
HORIZONTAL BLANKING INTERVAL

NOTES

1 In non interlaced operation the even field is repeated continuously, and the odd field is not

2 In interlaced operation the even field alternates with the odd field

Figure 21

JANUARY 1982

VIDEO ATTRIBUTES CONTROLLER (VAC) SC2673

Preliminary

DESCRIPTION

The Signetics 2673A and 2673B Video Attributes Controllers (VAC) are bipolar LSI devices designed for CRT terminals and display systems that employ raster scan techniques. Each contains a high speed video shift register, field and character attributes logic, attribute latch, cursor format logic and half dot shift control.

The VAC provides control of visual attributes on a field or character by character. Internal logic preserves field attribute data from character row to character row so that an attribute byte is not required at the beginning of each row. The 2673B provides for reverse video, blank (non-display), blink, underline and highlight attributes and a graphic smode attribute to work in conjunction with the Signetics 2670 Display Character and Graphics Generator (DCGG). The 2673A substitutes a light pen (strike-thru) attribute for the graphics attribute.

The horizontal dot frequency is the basic timing input to the VAC. Internally, this clock is divided down to provide a character clock output for system synchronization. Up to ten bits of video dot data are parallel loaded into the video shift register on each character boundary. The video data is shifted out on three outputs at the dot frequency. On the VIDEO output, the data is presented as a three level signal representing low, medium and high intensities. The three intensities are also encoded on two TTL compatible video outputs. Light or dark screen background can be selected.

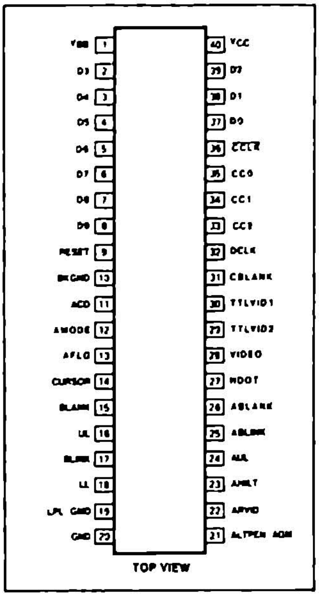
FEATURES

- 25MHz video dot rate
- Three level current driven (75 ohms) video output
- Three level encoded TTL video outputs
- Character/field attribute logic
  - Reverse video
  - Character blank
  - Character blink
  - Underline
  - Highlight
  - Light pen strike-thru or graphics control
- Field attributes extend from row to row
- Light or dark field
- Cursor reverse video logic
- Up to 10 dots per character
- Composite blanking for light field retrace
- Optional field graphics control output
- High speed bipolar design
- 40 pin dual in-line package
- TTL compatible
- Compatible with Signetics 2672 PVTC and 2670 DCGG

APPLICATIONS

- CRT terminals
- Word processing systems
- Small business computers

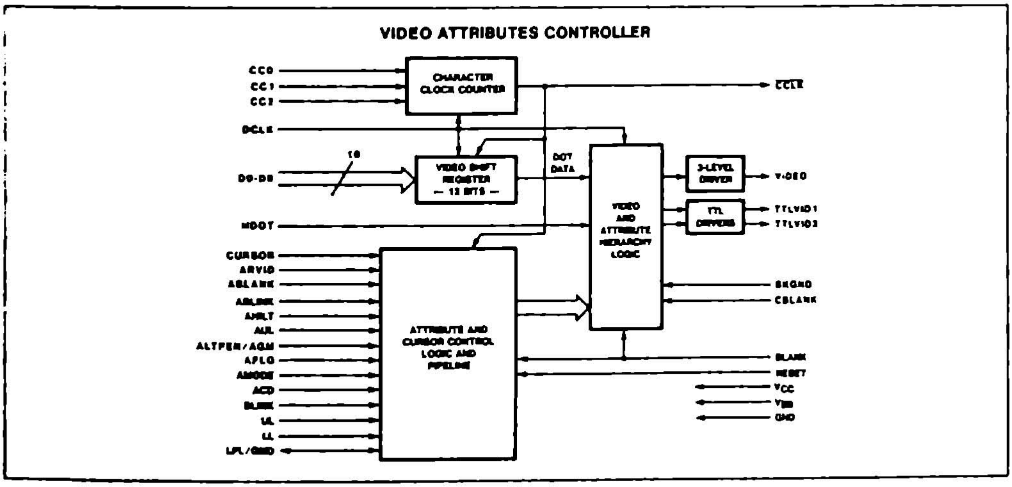
PIN CONFIGURATION


VBB 1
D3 2
D4 3
D5 4
D6 5
D7 6
D8 7
D9 8
RESET 9
BKGND 10
ACD 11
AMODE 12
AFLG 13
CURSOR 14
BLANK 15
UL 16
BLINK 17
LL 18
LPL GND 19
GND 20
40 VCC
39 D2
38 D1
37 D0
36 CCLK
35 CC0
34 CC1
33 CC2
32 DCLK
31 CBLANK
30 TTLVID1
29 TTLVID2
28 VIDEO
27 HDOT
26 ABLANK
25 ABLINK
24 AUL
23 AHILT
22 ARVID
21 ALTPEN AGM
TOP VIEW

ORDERING CODE

PACKAGES	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$, $V_{BB} = 1.5V \pm 10\%$, $T_A = 0°C$ to $70°C$	
	GRAPHICS ATTRIBUTE	LIGHT PEN ATTRIBUTE
Ceramic DIP	SC2673BC5I40	SC2673AC5I40
Plastic DIP	SC2673BC5N40	SC2673AC5N40

BLOCK DIAGRAM


VIDEO ATTRIBUTES CONTROLLER
CC0
CC1
CC2
DCLK
D0-D9
10
HDOT
CURSOR
ARVID
ABLANK
ABLINK
AHILT
AUL
ALTPEN/AGM
AFLG
AMODE
ACD
BLINK
UL
LL
LPL/GND
CHARACTER CLOCK COUNTER
VIDEO SHIFT REGISTER — 12 BITS —
DOT DATA
VIDEO AND ATTRIBUTE HIERARCHY LOGIC
ATTRIBUTE AND CURSOR CONTROL LOGIC AND PIPELINE
3-LEVEL DRIVER
TTL DRIVERS
CCLK
VIDEO
TTLVID1
TTLVID2
BKGND
CBLANK
BLANK
RESET
VCC
VBB
GND

VALVO

VIDEO ATTRIBUTES CONTROLLER (VAC) SC2673

Preliminary

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DCLK	32	I	**Dot Clock:** Dot frequency input. Video output shift rate.
$\overline{\text{CCLK}}$	36	O	**Character Clock:** A submultiple of DCLK. The frequency ranges from one sixth to one twelfth of DCLK, as determined by the state of the CC0-CC2 inputs.
CC2-CC0	33-35	I	**Character Clock Control:** The logic state on these three static inputs determine the internal divide factor for the $\overline{\text{CCLK}}$ output rate. Character clock rates of 6 thru 12 dots per character may be specified.
D0-D9	37-39, 2-8	I	**Dot Data Input:** These are parallel inputs corresponding to the character / graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the falling edge of each character clock.
HDOT	27	I	**Half Dot Shift:** When this input is high, the serial video output is delayed by one half dot time. This input is latched on the falling edge of each character clock.
CURSOR	14	I	**Cursor Timing:** This input provides the timing for the cursor video. When high, effectively reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.
BKGND	10	I	**Background Intensity:** Specifies light or dark video during BLANK and character fields. Affects the intensities of all attributes.
BLANK	15	I	**Screen Blank:** When high, this input forces the video outputs to the level specified by the BKGND input (either high or low intensity). Not effective when CBLANK is high.
CBLANK	31	I	**Composite Blank:** Used with the TTL video outputs only. When high, this input forces the video outputs to a low intensity state for retrace blanking. When BKGND input is low, or when using video outputs, this input may be tied low.
ARVID	22	I	**Reverse Video Attribute:** The intensity of the associated character or field video is reversed. All other attributes are effectively reversed.
AHILT	23	I	**Highlight Attribute:** All dot video (including underline) of the associated character or field is highlighted with respect to the BKGND input and the reverse video attribute.
ABLANK	26	I	**Blank Attribute:** Generates a blank space in the associated character or field. The blank space intensity is determined by the BKGND input, the reverse video attribute, and the CURSOR input.
ABLINK	25	I	**Blink Attribute:** The associated character or field video is driven to the intensity determined by BKGND and the reverse video attribute when the BLINK input is high.
AUL	24	I	**Underline Attribute:** Specifies a line to be displayed on the character or field. The line is specified by the UL input. All other attributes apply to the underline video.
ALTPEN / AGM	21	I	**Light Pen Attribute (2673A):** Specifies a highlighted line to be displayed on the character or field. The line is specified by the LPL input.
	•	I	**Attribute Graphics Mode (2673):** This input is latched and synchronized to provide a field GMD output for the 2670 DCGG.
AMODE	12	I	**Attribute Mode:** Specifies character (AMODE = 0) or field (AMODE = 1) attributes mode.
AFLG	13	I	**Attributes Flag:** The VAC samples and latches the attributes inputs when this input is high. If field attributes are specified (AMODE = 1), the attributes are double buffered on a row basis. Thus, each scan line of every character row will start with the attributes that were valid at the end of the previous row.
ACD	11	I	**Attribute Control Display:** In field attributes mode (AMODE = 1), if ACD = 0, the first character in each new attribute field (the attribute control character) will be suppressed and only the attributes will be displayed. If ACD = 1, the first character and the attributes are displayed. This input has no effect in character mode (AMODE = 0).

VIDEO ATTRIBUTES CONTROLLER (VAC) SC2673

Preliminary

PIN DESIGNATION (continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
BLINK	17	I	**Blink:** This input is sampled on the falling edge of BLANK to provide the blink rate for the character blink attribute. It should be a submultiple of the frame rate.
UL	16	I	**Underline:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.
LPL/GMD	19	I	**Light Pen Line (2673A):** Indicates the scan line(s) for the light pen strike-thru attribute. Latched on the falling edge of BLANK
		O	**Graphics Mode (2673):** This output provides a synchronized, latched, field graphics mode corresponding to the AGM input. This output can be used to control the GM input on the 2670 DCGG.
LL	18	I	**Last Line:** Indicates the last scan line of each character row. Used internally to extend field attributes across row boundaries. Latched on the falling edge of BLANK. This input has no effect in character mode (AMODE = 0).
VIDEO	28	O	**Video:** A three level serial video output which corresponds to the composite dot pattern of characters, attributes and cursor.
TTLVID1	30	O	**TTL Video 1:** This output corresponds to the serial, non-highlighted video dot pattern.
TTLVID2	29	O	**TTL Video 2:** This output corresponds to the highlighted serial video dot pattern. Should be used with TTLVID1 to decode a composite video of three intensities.
RESET	9	I	**Manual Reset:** This active high input initializes the internal logic and resets the attribute latches. Normally used for testing.
V_{CC}	40	I	**Power Supply:** +5 Volts ± 5%
V_{BB}	1	I	**Bias Supply:** +1.5 Volts ± 10%
GND	20	I	**Ground:** 0V reference

FUNCTIONAL DESCRIPTION

The VAC consists of four major sections (see block diagram). The high speed dot clock input is divided internally to provide a character clock for system timing. The parallel dot data is loaded into the video shift register on each character boundary and shifted into the video logic block at the dot rate. The six attribute inputs are latched internally and combined with the serial dot data to provide a three level video source for the monitor.

A separate BLANK input defines the active screen area. When BLANK = 0, the video levels are derived internally by the combinations of dot data, attributes, cursor, and the state of the BKGND input. Either black or white background can be selected. Symbols (dot data) are normally gray and can be highlighted to white or black as shown in figure 1. Note that the VIDEO output is inverted as referenced to the TTL video outputs.

During the inactive screen area (BLANK = 1) the video level produced by the TTL outputs is either white (BKGND = 1) or black (BKGND = 0). A separate composite blank (CBLANK) input is provided to suppress raster retrace video when white background is specified. During the inactive screen area (BLANK = 1), the video level produced by the VIDEO output is either black (BKGND = 1) or white (BKGND = 0). For the latter case, raster retrace video suppression is accomplished by raising the BKGND input during horizontal and vertical retrace intervals. For black background, tie BKGND high. Tie CBLANK input low for both cases.

Character Clock Counter

The character clock counter divides the frequency on the DCLK input to generate the character clock ($\overline{CCLK}$). The divide factor is specified by the clock control inputs (CC0-CC2) as follows:

CC2	CC1	CC0	$\overline{CCLK}$	
			DOTS/CHARACTER	DUTY CYCLE
0	0	0	6	3/3
0	0	1	6	3/3
0	1	0	7	4/3
0	1	1	8	4/4
1	0	0	9	5/4
1	0	1	10	5/5
1	1	0	11	6/5
1	1	1	12	6/6

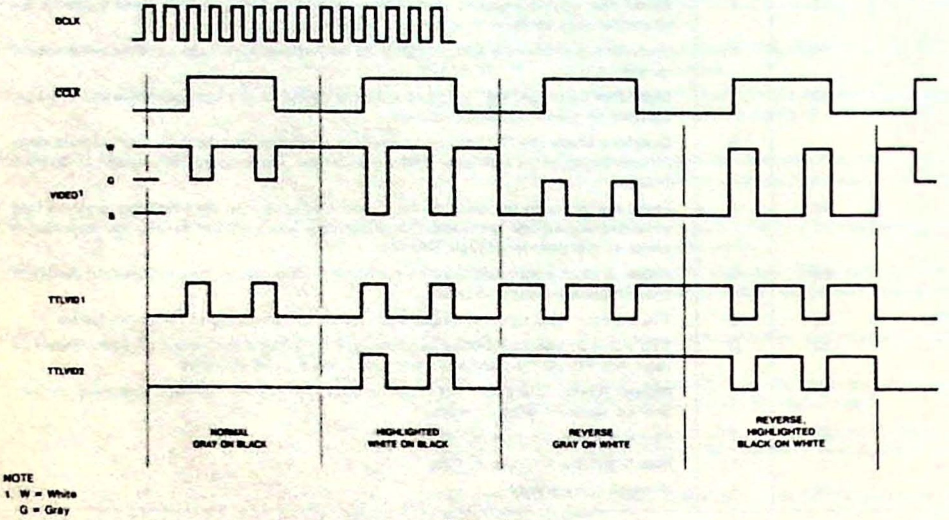
Video Shift Register

On each character boundary, the parallel data (D0-D9) is loaded into the video shift register. The data is shifted out least significant bit first (D0) by the DCLK. If 11 or 12 dots/character are specified (CC2-CC0 = 110 or 111), a 0 (blank dot) is always shifted out before D0. For 12 dots/character, a 0 is also shifted out after D9. The serial dot data is shifted into the video logic where it is combined with the cursor and attributes to encode three levels of video.

Attribute and Cursor Control

The VAC visual attributes capabilities include: reverse video, character blank, blink, underline, highlight, and light pen strike-thru. The six attributes and the three attribute control inputs (AMODE, AFLG, and ACD) are clocked into the VAC on the falling edge of $\overline{CCLK}$. If AFLG is high, the attributes are latched internally and are effective for either one character time (AMODE = 0) or until another set of attributes is latched (AMODE = 1). The attributes set is double buffered on a row by row basis internally. Using this technique, field attributes can extend across character row boundaries thereby

ENCODED VIDEO OUTPUTS



NOTE
1. W = White
G = Gray
B = Black

Figure 1

TTLVID2	TTLVID1	INTENSITY
0	0	Black (or CBLANK)
0	1	Gray (on black surround)
1	0	Gray (on white surround)
1	1	White

NOTE
The TTLVID1 output can be used independently to generate a two level non-highlighted video

eliminating the necessity of starting each row with an attribute set.

When field attribute mode is selected, (AMODE = 1), the VAC will accommodate two attribute storage configurations. In one configuration, the attribute control data is stored in the refresh RAM, taking the place of the first character code in the field to be affected. For this mode, the ACD input is tied low and blank characters will be displayed in the screen positions occupied by the attribute data (see figure 10). In the second configuration, (ACD = 1), the character codes and attribute data are presented to the VAC in parallel. In this mode, dot data is displayed at each character position (see figure 11).

The CURSOR and the attribute input signals are pipelined internally to allow for system propagations (one $\overline{CCLK}$ for refresh RAM, one $\overline{CCLK}$ for dot generator). The attribute timing signals BLINK, UL, LPL and LL are clocked into the VAC at the beginning of each scan line by the falling edge of the BLANK input. Thus, these signals must be in their proper state at the falling edge of BLANK preceding the scan line at which they are to be active (see figure 4).

Video Logic

The serial dot data and the pipelined cursor and attributes are combined to generate the three level current source on the VIDEO output. The three levels (white, gray, and black) are also encoded on the two TTL compatible outputs TTLVID1 and TTLVID2. The three levels are encoded as shown.

The video is normally shifted out on the leading edge of the DCLK. When the HDOT input is asserted, the corresponding dot data is delayed by one-half DCLK. This half dot shifting, when used on selected lines of character video, can be used to effect eye-pleasing character rounding as shown in figure 2.

Attribute Hierarchy

The video of each character block consists of four components as shown in figure 3.

Symbol video is generated from the dot data inputs D0-D9.

Underline video is enabled by the AUL attribute and is generated when the UL timing input is active. Underline and symbol video are always the same intensity.

Strike-thru video is enabled by the ALTPEN attribute and is generated when the LPL timing input is active. This video is always highlighted and takes precedence over the symbol and underline video. This feature applies to the 2673A only.

Surround video is the absence of symbol, underline and strike-thru video or the presence of the non-display attributes (ABLANK or ABLINK • BLINK).

The relative intensities of the four video components are determined by the remaining attributes (AHILT, ABLANK, ABLINK, ARVID) and the BKGND and CURSOR inputs as illustrated in table 1.

"AT" SYMBOL WITH AND WITHOUT HALF DOT SHIFTING

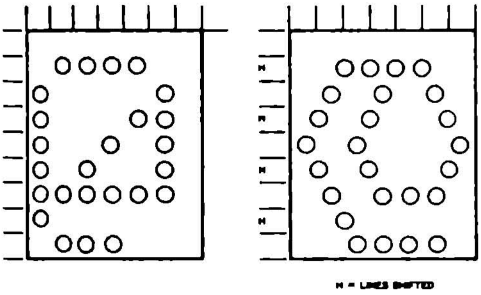


Figure 2

VIDEO COMPONENTS OF CHARACTER BLOCK

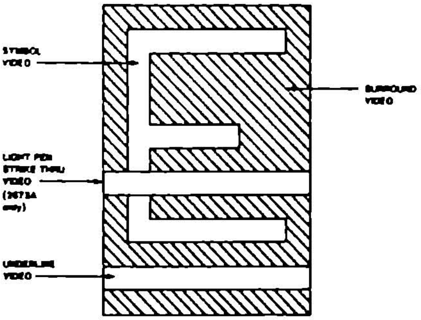


Figure 3

ATTRIBUTES AND CONTROL INPUTS d = don't care				RELATIVE VIDEO INTENSITIES W = White, B = Black, G = Gray		
BKGND[5]	REVERSE[1]	NON-DISPLAY[2]	AHILT	STRIKE THRU VIDEO[3]	SYMBOL OR UNDERLINE VIDEO[3,4]	SURROUND VIDEO[3]
0	0	0	0	W	G	B
0	0	0	1	W	W	B
0	0	1	d	B	B	B
0	1	0	0	B	G	W
0	1	0	1	B	B	W
0	1	1	d	W	W	W
1	0	0	0	B	G	W
1	0	0	1	B	B	W
1	0	1	d	W	W	W
1	1	0	0	W	G	B
1	1	0	1	W	W	B
1	1	1	d	B	B	B

NOTES
1 Reverse = $\text{ARVID} \cdot \overline{\text{CURSOR}} + \overline{\text{ARVID}} \cdot \text{CURSOR}$
2 Non-display = ABLANK + ABLINK • BLINK
3 See figure 3
4 Symbol and underline video are always the same intensity
5 Reverse sense for VIDEO output

Table 1. ATTRIBUTES HIERARCHY

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground	−0.5 to +6.0	V

NOTES
1 Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2 For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (ceramic package).

VIDEO ATTRIBUTES CONTROLLER (VAC) SC2673

Preliminary

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, $V_{BB} = +1.5V \pm 10\%^{3,4,5}$

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage (except VIDEO)	$I_{OL} = 1.6mA$			0.4	V
V_{OH}	Output high voltage (except VIDEO)	$I_{OH} = -100\mu A$	2.4			V
V_B	VIDEO black level	$R_L = 75\Omega$ to GND		0		V
V_G	VIDEO gray level	$R_L = 75\Omega$ to GND		0.225		V
V_W	VIDEO white level	$R_L = 75\Omega$ to GND		0.50		V
I_{IL}	Input low current	$V_{IN} = 0.4V$			$-400/-800^{10}$	μA
I_{IH}	Input high current	$V_{IN} = 2.4V$			$20/40^{10}$	μA
I_{CC}	V_{CC} supply current	$V_{IN} = 0V$, $V_{CC} = max$			50	mA
I_{BB}	V_{BB} supply current	$V_{BB} = max$			100	mA

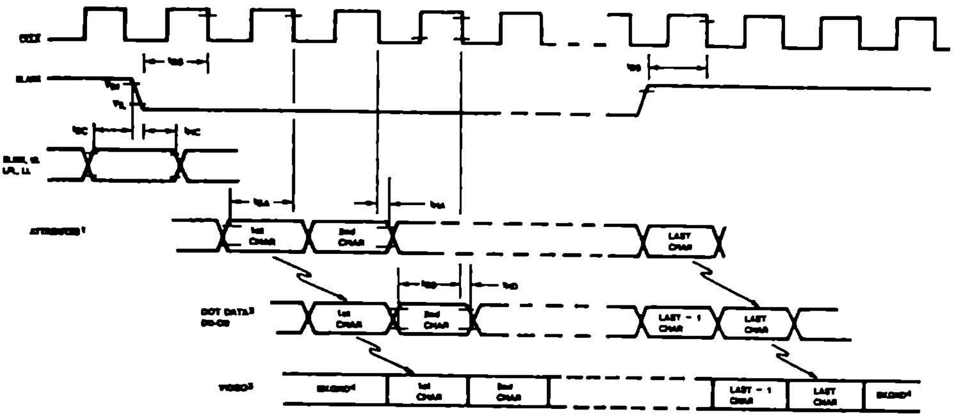
AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, $V_{BB} = +1.5V \pm 10\%^{3,4,5}$

PARAMETER		TEST CONDITIONS	TENTATIVE LIMITS			UNIT
			Min	Typ	Max	
Dot clock[6]						
t_D	frequency				25	MHz
t_{DH}	high		15			ns
t_{DL}	low		15			ns
Setup times to $\overline{\text{CCLK}}$[7]						
t_{BS}	BLANK		35			ns
t_{SC}	BLINK, UL, LPL, LL (ref to BLANK)		20			ns
t_{SA}	Attributes		35			ns
t_{SD}	Dot data D0-D9		70			ns
t_{SK}	CURSOR		35			ns
t_{FS}	AFLG		50			ns
t_{SH}	HDOT		35			ns
Hold times from $\overline{\text{CCLK}}$[7]						
t_{HC}	BLINK, UL, LPL, LL (ref to BLANK)		20			ns
t_{HA}	Attributes		35			ns
t_{HD}	Dot data D0-D9		0			ns
t_{HK}	CURSOR		35			ns
t_{FH}	AFLG		50			ns
t_{HH}	HDOT		35			ns
Setup times to DCLK[8]						
t_{SG}	BKGND		15			ns
t_{SB}	CBLANK		15			ns
Hold times from DCLK[8]						
t_{HG}	BKGND		15			ns
t_{HB}	CBLANK		15			ns
Delay times[9]		$C_L = 150pF$				
t_{DGM}	GMD from DCLK				60	ns
t_{DC}	CCLK from DCLK				80	ns
t_{DV}	VIDEO, TTLVID1, and TTLVID2 from DCLK				75	ns

NOTES

3. Parameters are valid over operating temperature range unless otherwise specified.
4. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V. All time measurements are referenced at input voltages of 0.8V, 2.0V and at output voltages of 0.8V, 2.0V as appropriate.
5. Typical values are at +25°C, typical supply voltages and typical processing parameters.
6. See figure 7. Half dot shift feature 18MHz max.
7. See figures 4, 5, 6, and 9.
8. See figure 8.
9. See figures 6 and 7.
10. For DCLK input.

VAC PIPELINE TIMING


DCLK
BLANK
t_{BD}
V_{IH}
V_{IL}
t_{SC}
t_{HC}
ATTRIBUTES¹
1st CHAR
2nd CHAR
LAST CHAR
DOT DATA² D0-D7
LAST - 1 CHAR
VIDEO³
BKGND⁴

NOTES
1 Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, and ALTPEN.
2 One $\overline{DCLK}$ delay for dot data (obtained from delay through character generator).
3 See figure 7 for detail timing of VIDEO, TTLVID1, TTLVID2.
4 Non-active scan time. VIDEO reverts to polarity selected by the BKGND input.

Figure 4

CURSOR PIPELINE TIMING

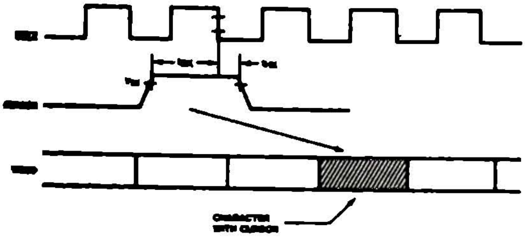
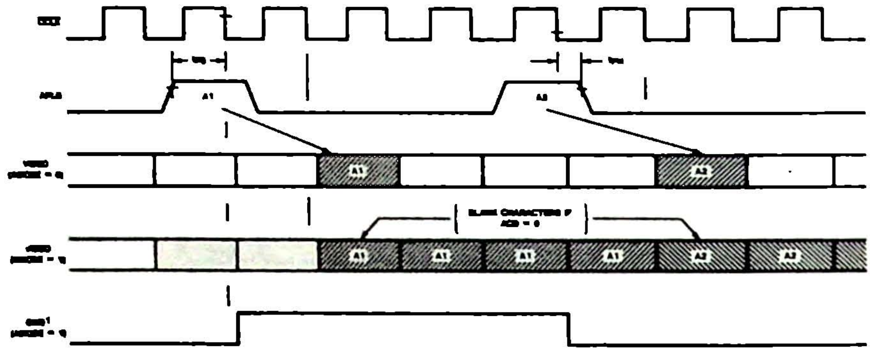

DCLK
t_{SC}
t_{HC}
V_{IH}
CURSOR
VIDEO
CHARACTER WITH CURSOR

Figure 5

Preliminary

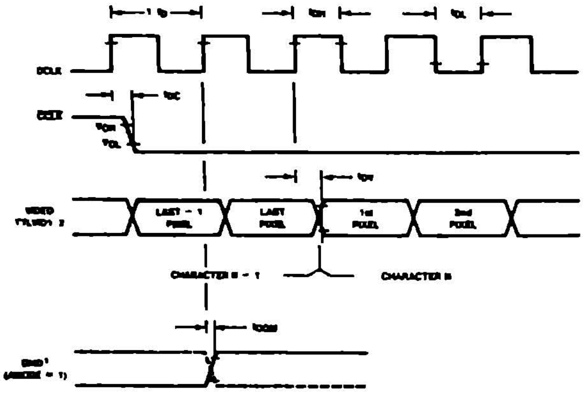
CHARACTER (AMODE = 0), FIELD (AMODE = 1), AND GMD ATTRIBUTE TIMING


CCLK
A1
A2
VIDEO (AMODE = 0)
A1
A2
BLANK CHARACTERS IF ACB = 0
VIDEO (AMODE = 1)
A1
A2
GMD¹ (AMODE = 1)

NOTE
1. GMD output in 2673 version only. See figure 7 for detail timing

Figure 6

VIDEO AND GMD PIPELINE TIMING


1 tp
tCH
tCL
DCLK
tDC
CCLK
tCH
tCL
tDV
VIDEO (TTLVID1, 2)
LAST − 1 PIXEL
LAST PIXEL
1st PIXEL
2nd PIXEL
CHARACTER n − 1
CHARACTER n
tDGM
GMD¹ (AMODE = 1)

NOTE
1. GMD output in 2673 version only

Figure 7

BKGND AND CBLANK TIMING DURING INACTIVE SCAN TIME (BLANK = 1)

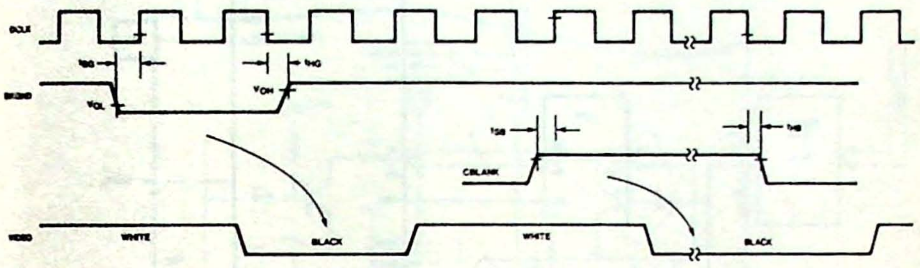

DCLK
tSG
tHG
BKGND
VOL
VOH
tSB
tHB
CBLANK
VIDEO
WHITE
BLACK
WHITE
BLACK

Figure 8

HALF DOT SHIFT TIMING

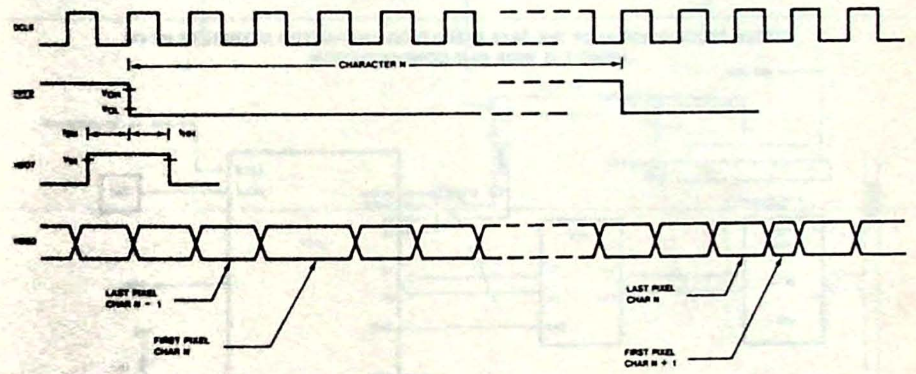

DCLK
CHARACTER N
VOH
VOL
VOH
VIDEO
LAST PIXEL CHAR N − 1
FIRST PIXEL CHAR N
LAST PIXEL CHAR N
FIRST PIXEL CHAR N + 1

Figure 9

SYSTEM BLOCK DIAGRAM OF THE 2673 IN FIELD ATTRIBUTE MODE USING THE NARROW RAM (8 WIDE) CONFIGURATION

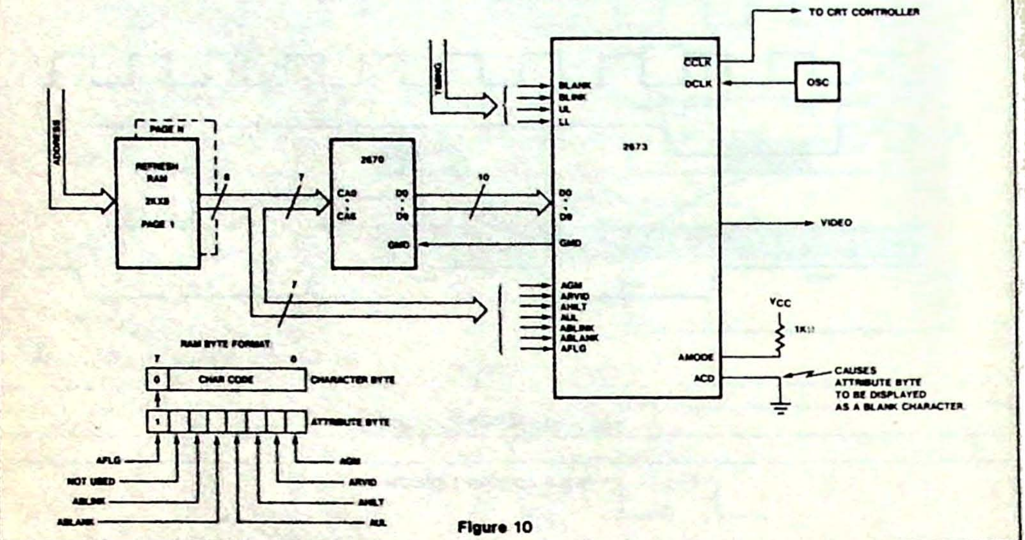
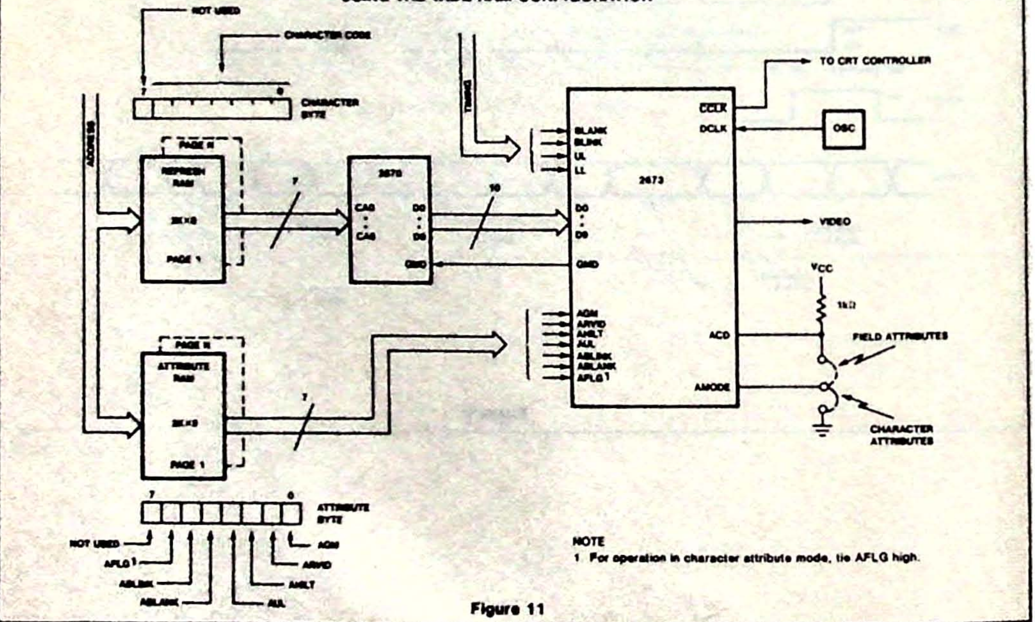

TO CRT CONTROLLER
TIMING
ADDRESS
PAGE N
REFRESH RAM
2K×8
PAGE 1
8
7
2670
CA0
CA6
D0
D9
GND
10
7
BLANK
BLINK
UL
LL
2673
CCLK
DCLK
OSC
D0
D9
GND
VIDEO
AGM
ARVID
AHLT
AUL
ABLINK
ABLANK
AFLG
VCC
1KΩ
AMODE
ACD
CAUSES ATTRIBUTE BYTE TO BE DISPLAYED AS A BLANK CHARACTER
RAM BYTE FORMAT
7
0
0
CHAR CODE
CHARACTER BYTE
1
ATTRIBUTE BYTE
AFLG
NOT USED
ABLINK
ABLANK
AGM
ARVID
AHLT
AUL

Figure 10

SYSTEM BLOCK DIAGRAM OF THE 2673 IN FIELD OR CHARACTER ATTRIBUTE MODE USING THE WIDE RAM CONFIGURATION


NOT USED
CHARACTER CODE
7
0
CHARACTER BYTE
ADDRESS
TIMING
TO CRT CONTROLLER
PAGE N
REFRESH RAM
2K×8
PAGE 1
7
2670
CA0
CA6
D0
D9
GND
10
BLANK
BLINK
UL
LL
2673
CCLK
DCLK
OSC
D0
D9
GND
VIDEO
VCC
1kΩ
AGM
ARVID
AHLT
AUL
ABLINK
ABLANK
AFLG1
ACD
AMODE
FIELD ATTRIBUTES
CHARACTER ATTRIBUTES
PAGE N
ATTRIBUTE RAM
2K×8
PAGE 1
7
7
0
ATTRIBUTE BYTE
NOT USED
AFLG1
ABLINK
ABLANK
AGM
ARVID
AHLT
AUL

NOTE
1. For operation in character attribute mode, tie AFLG high.

Figure 11

VIDEO OUTPUT STAGES OF THE 2673

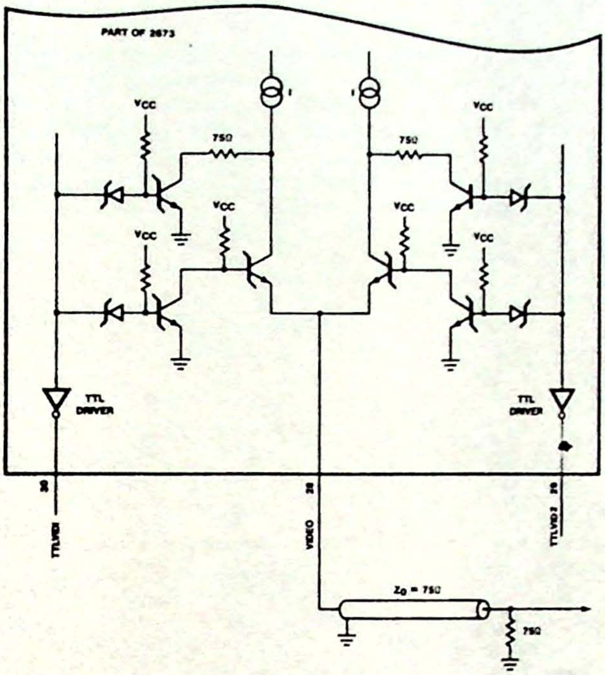

PART OF 2673
I
I
VCC
VCC
VCC
VCC
VCC
VCC
75Ω
75Ω
TTL DRIVER
TTL DRIVER
30
TTLVID1
28
VIDEO
26
TTLVID2
Z0 = 75Ω
75Ω

Figure 12

Section 3
Single Chip
8-Bit Microcomputers

VALVO

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The MAB8021 is a single-chip 8-bit microcomputer that is fabricated using the N-MOS silicon gate process. It contains a 1K x 8 program memory, a 64 x 8 data memory, 21 I/O lines, and an 8-bit timer/event counter, in addition to on-board oscillator and clock circuits. For systems that require extra I/O capacity, the MAB8021 can be expanded using the 8243 or discrete logic.

This microcomputer is designed to be an efficient controller as well as an arithmetic computer. The MAB8021 has bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set (see Table 4) consisting mostly of single byte instructions and no instructions over two bytes in length.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead package
- 1K x 8 ROM, 64 x 8 RAM, 21 I/O lines
- Internal timer/event counter
- Clock generated with single inductor or crystal
- Single 5 V supply (range: + 4,5 V to + 6,5 V)
- 8,38 µs cycle time; all instructions 1 or 2 cycles
- Instructions: MAB8048 subset
- Zero-cross detection capability
- Easily expandable I/O

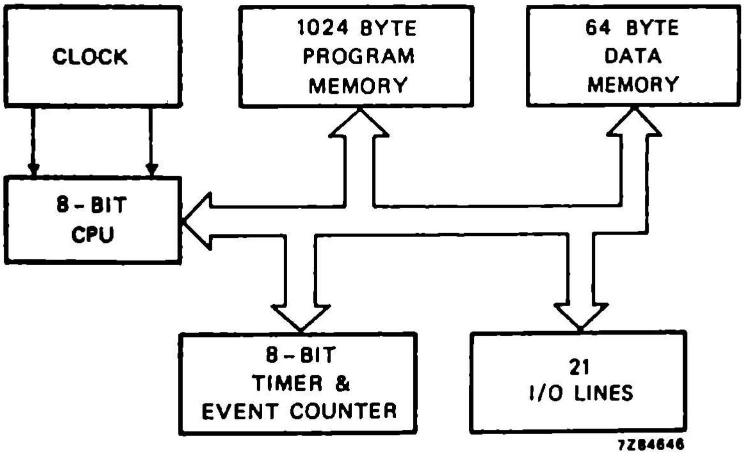


Fig. 1 Block diagram.

PACKAGE OUTLINES

MAB 8021 P: 28-lead DIL; plastic (SOT-117).
MAB 8021 D: 28-lead DIL; ceramic (SOT-135).
MAB 8021 T: 28-lead flat-pack; plastic (SO-28, SOT-136A).

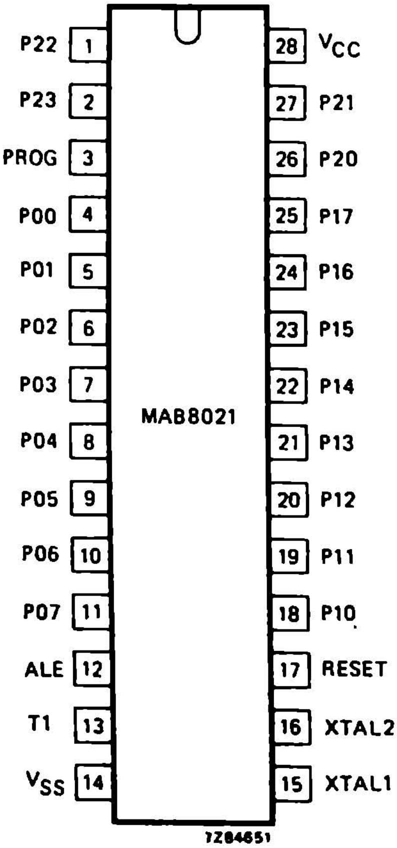

P22
P23
PROG
P00
P01
P02
P03
P04
P05
P06
P07
ALE
T1
V_{SS}
V_{CC}
P21
P20
P17
P16
P15
P14
P13
P12
P11
P10
RESET
XTAL2
XTAL1
MAB8021
7Z84651

Fig. 2 Pinning diagram.

PIN DESIGNATION

designation	pin no.	function
V_{CC}	28	Power supply: + 5,5 V.
V_{SS}	14	Ground.
PROG	3	Output strobe: for 8243 I/O expander.
P00-P07	4-11	Port 0: 8-bit quasi-bidirectional port.
P10-P17	18-25	Port 1: 8-bit quasi-bidirectional port.
P20-P23	26,27,1,2	Port 2: 4-bit quasi-bidirectional port. P20-P23 also serve as a 4-bit I/O expander bus for the 8243.
T1	13	T1: input pin testable using the JT1 and JNT1 instructions. Can be designated the timer/event counter input, using the STRT CNT instructions. Also allows zero-crossover sensing of slowly moving a.c. inputs.
RESET	17	Reset: input, used to initialize the processor by clearing status flip-flops and setting program counter to zero.
ALE	12	Address latch enable: signal occurring once every 30 input clocks, used as an output clock.
XTAL1	15	Timing control element: one side of crystal or inductor input for internal oscillator. Also the input for an external source (not TTL compatible).
XTAL2	16	Timing control element: other side.

FUNCTIONAL DESCRIPTION

The following is a functional description of the MAB8021.

Program memory

The program memory consists of a 1024 byte mask-programmed ROM. No external expansion capability is provided. The first instruction must be at location 0.

Page organization

The program memory is organized as four pages of 256 bytes each. Only the unconditional branch instruction (JMP) can cause jump over page boundaries. The CALL instruction can transfer control to a subroutine on any page. Likewise, the table data reference instruction (MOVP A,@,A) can only access data located on the same page as the instruction.

Data memory

The 64 byte dynamic RAM is organized as shown in Fig. 3. Locations (registers) 0 to 7 are directly addressable by using several instructions. All locations are indirectly addressable by using registers R0 or R1.

Address stack

Locations 8 to 23 are used as the address stack. The address stack enables the processor to keep track of the return addresses generated from CALL instructions. A 3-bit stack pointer (SP) supplies the address of the locations to be loaded with the next return address generated. The SP to this push-down stack is incremented by one after a return address is stored, and decremented by one before an address is fetched during a RET (return instruction). A total of 8 levels of nesting is possible. The SP is initialized to location 8 upon RESET.
Since each address is 10-bits long, two bytes must be used to store a single address. The SP is incremented and decremented by one, but each increment or decrement moves the address pointed to, by two. Therefore, only even numbered addresses are pointed to. If a particular application does not require 8 levels of nesting, the unused portion of the stack may be used as any other indirectly addressable scratchpad location.

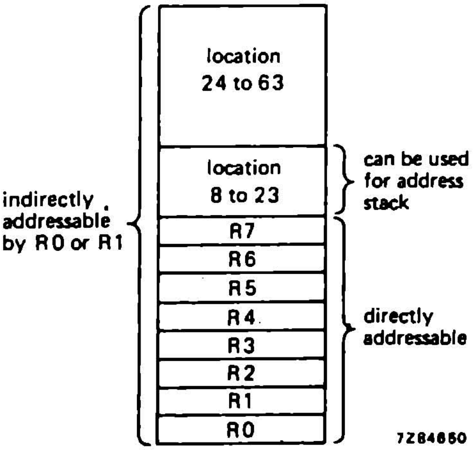


Fig. 3 Internal RAM memory map.

FUNCTIONAL DESCRIPTION (continued)

Program variable storage

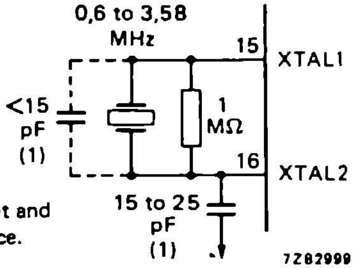
Locations 24 to 63 may be used for storage of program variables or data. In addition, any portion of the stack area (locations 8 to 23) not used for return address storage may also be used for variable storage.

Oscillator and clock

The MAB8021 contains its own internal oscillator and clock driver. The frequency is determined by a single crystal, or inductor. All internal time slots are derived from the external element, and all outputs are a function of the oscillator frequency. The particular control element is connected between pins 15 (XTAL1) and 16 (XTAL2).

An instruction cycle consists of 10 states, and each state is a time slot of 3 oscillator periods. Therefore, to obtain a 10 μs instruction cycle, a 3 MHz crystal should be used.

The MAB8021 utilizes dynamic RAM and certain other dynamic logic. Due to the clocking required with dynamic circuits, the oscillator frequency must be equal to or greater than 600 kHz, or improper operation may occur.



(1) Including socket and stray capacitance.

Counter/timer

The MAB8021 has an internal counter that can, under program control, count either external events (T1, pin 13) or instruction cycles. The instructions that control the counter and the functions they perform are summarized in Table 1.

The counter is an 8-bit binary up-counter. When used as a timer, the input to the counter is the overflow of a 5-bit (÷ 32) prescaler. The input of the prescaler is a signal that occurs each instruction cycle (30 clock periods).

When used as a counter, HIGH-to-LOW transitions on the T1 input are counted. The maximum frequency that can be counted is one third of the instruction cycle frequency (33,3 kHz for a 3 MHz oscillator). The positive duration of the signal must be a minimum of one tenth of the period of the instruction cycle.

When the 8-bit counter overflows, a flag is set. The flag can be tested using the JTF (jump if timer flag = 1) instruction, which resets the flag.

Table 1 Counter/timer control

function	used together counts instruction cycles		counter
	prescaler	timer	
CLEAR	STRT T	MOV T,A@(A) = 0	MOV T,A@(A) = 0
PRESET	–	MOV T,A	MOV T,A
START	STRT T	STRT T	STRT CNT
STOP	STOP TCNT	STOP TCNT or RESET	STOP TCNT or RESET
TEST	–	JTF	JTF
READ*	–	MOV A,T	MOV A,T

* READ does not disturb the counting process.

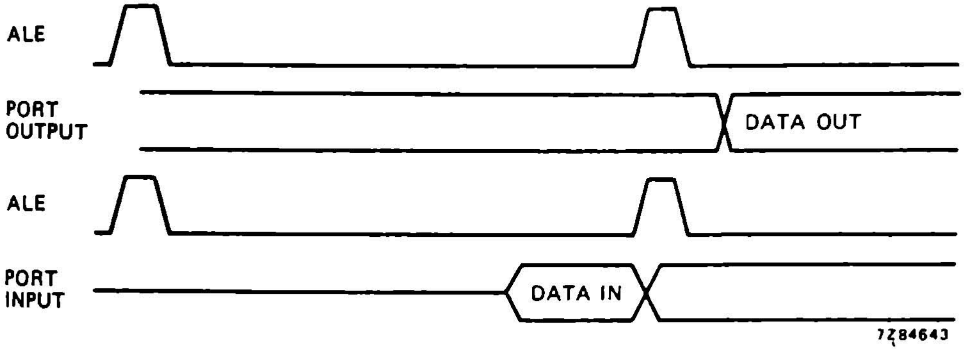


Fig. 4 Port 2 timing; normal operation.

Input/output

The 21 I/O pins are arranged as two 8-bit quasi-bidirectional ports, one 4-bit quasi-bidirectional port, and the T1 input. The 20 port pins can be individually assigned, under program control, to be either inputs or outputs (at a given time) or both (at different times).

Port operation

A simplified representation of the port circuitry is shown in Fig. 5. Output data written to a port is latched and remains unchanged until rewritten. Writing a '1' to the ports turns on the large pull-up device for less than the instruction time, which reduces the rise time of the signal. The smaller pull-up device then maintains the '1' level. The on-resistance of the small pull-up is approximately 30 kΩ. This means that, when used as an input pin, the external circuitry must sink only 0,2 mA at V_{IL}, so the MAB8021 recognizes a '0'. Thus, writing a '1' to a port-pin, it is enabled to be used either as an input pin or as a true HIGH level, latched, output pin. Writing a '0' to a port, a large pull-down device is turned on, which is capable of sinking 1,6 mA into V_{SS}. The preceding applies to ports 1 and 2.

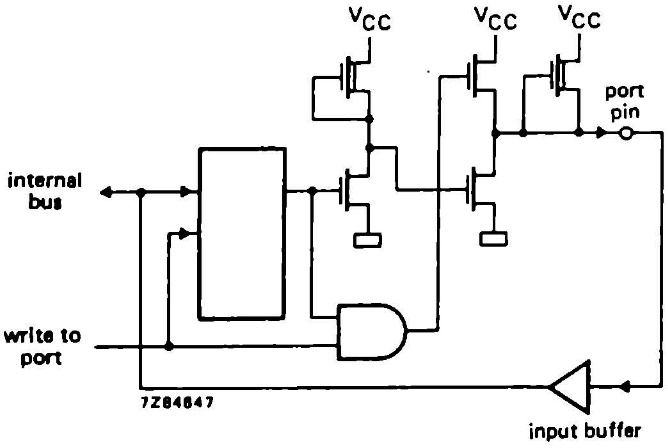


Fig. 5 Quasi-bidirectional port.

FUNCTIONAL DESCRIPTION (continued)

Port operation (continued)

Port 0 (P00 to P07) does not have the large pull-up device. In addition, by mask option, the small pull-up device (on any pin) can be deleted, thus providing a true open drain output.
The instructions ANL and ORL can be used to mask the ports, line-by-line, so that any combination of inputs and outputs can be configured on any port.

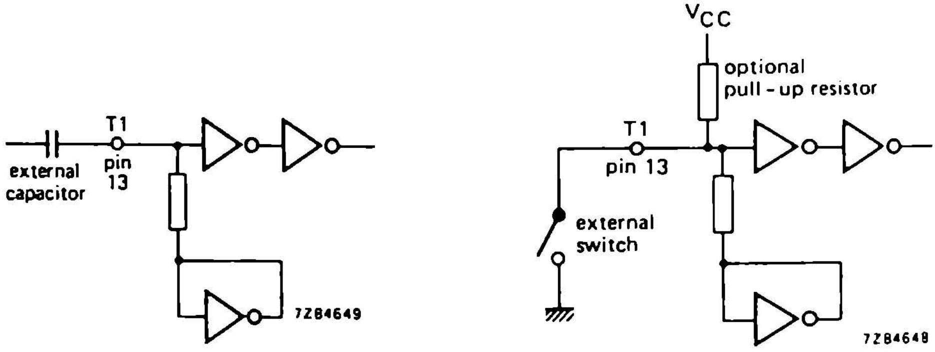
High-current outputs

Two pins (P10 and P11) are provided, which can sink larger currents (7 mA each, at V_{SS} + 2,5 V), when required. These pins may be used in parallel for 14 mA drive if the output logic states are always the same.

T1 input

The T1 input can be used for the following functions:
- event counter (external input)
- test input for branch instructions
- zero voltage cross-over detection.

The operation of T1 as an input to the event counter is described under the heading "Counter/timer". When used as a test input, the JT1 and JNT1 instructions test for '1' and '0' levels respectively.
The T1 pin can also be used to detect the zero-crossing of slowly moving a.c. signals (50 Hz). The self-biasing circuit shown in Fig. 6 permits the T1 input to detect when the input voltage crosses zero within ± 135 mV, when the voltage is coupled through a 0,2 µF capacitor. The maximum input voltage is 3 V peak-to-peak. The zero-cross detection is especially useful in thyristor control of 50 Hz power equipment and in developing time-of-day and other timing routines. A pull-up resistor can be provided as a ROM mask option. This is useful when the input is from a switch or a standard TTL output.



(a) (b)

Fig. 6 T1 pin options for (a) zero-cross applications and with (b) optional pull-up resistor.

Expanded I/O

The MAB8021 can be used with the 8243 I/O expander chip, which provides additional I/O capability. The 8243 has 4 directly addressable 4-bit ports. It connects the PROG pin, which provides the clock, and pins P20 to P23, which provide address and data. These ports can be written with a MOVD P,A, ANLD P,A, and ORLD P,A for ports 4 to 7. Reading is via MOVD A,P instruction. The previous data on P20 to P23, before an output expander instruction, is lost. Therefore, when using an output expander, P20 to P23 are not useful for general input/output operation. This circuit configuration is shown in Fig. 7 and the timing diagram in Fig. 8.
The MAB8021 can also use standard low-cost TTL circuits, to expand the number of I/O lines.

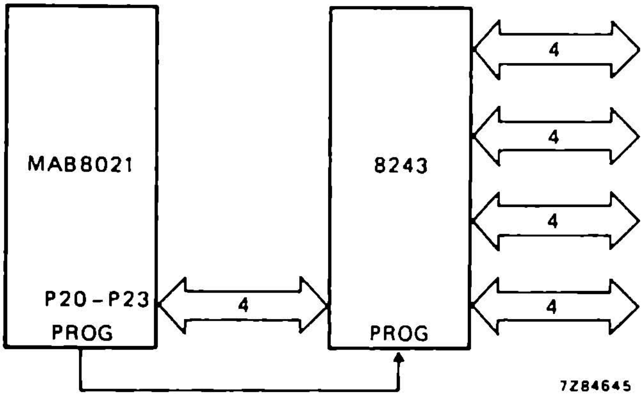

MAB8021
P20-P23
PROG
4
8243
PROG
4
4
4
4
7Z84645

Fig. 7 Expander interface.

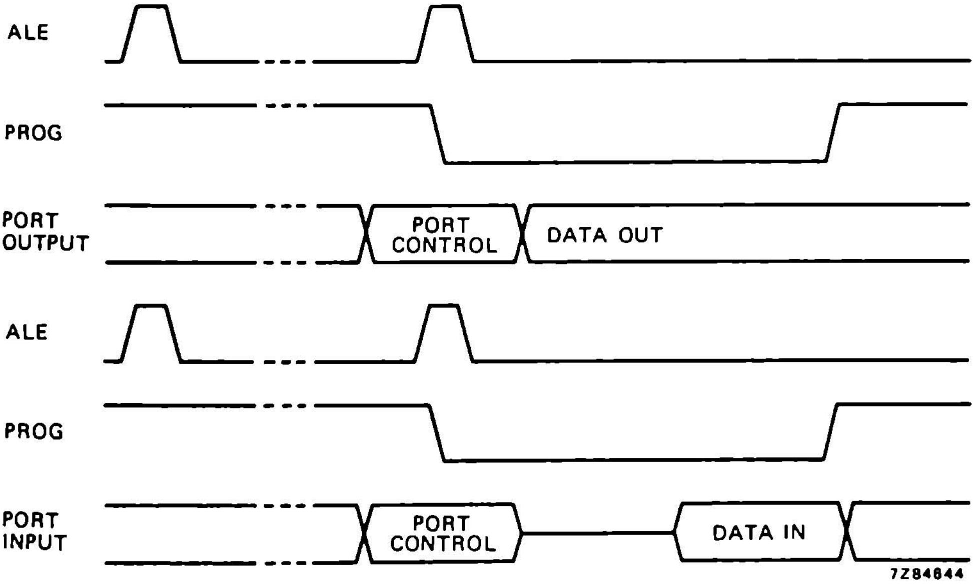

ALE
PROG
PORT OUTPUT
PORT CONTROL
DATA OUT
ALE
PROG
PORT INPUT
PORT CONTROL
DATA IN
7Z84644

Fig. 8 Port 2 timing; expander operation.

FUNCTIONAL DESCRIPTION (continued)

Central Processing Unit

The MAB8021 has arithmetic, logical and conditional branching capabilities. The DA A, SWAP A, and XCHD instructions simplify BCD arithmetic and handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the current ROM page.
The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. In addition, the DJNZ instruction decrements a designated register and branches if the contents are not zero. This instruction is useful for looping control.

Testing and debug

To facilitate testing and debug, certain test modes may be activated in the MAB8021 by raising combinations of RESET, T1 and PROG to 15 V. The internal ROM is dumped out sequentially for verification. External memory operation is used for CPU check-out.

RESET	PROG	T1	case	function
5 V	X	X		power-on clear
0 V	X	X		normal operation
15 V	15 V	⎍	mode 1b	On every T1 rising edge the program counter increments, dumps internal ROM to Port 0.
0 V	15 V	X	mode 2	IC will operate from external memory (one page) via Port 0. ALE strobes address out, memory in.
15 V	X	X	mode 3	IC accepts op-codes into Port 1; allows Port 0 and 8243 testing.

X = normal mode (between 0 V and V_{CC}).

Differences between the MAB8021 and the MAB8048

Although the MAB8021 is basically an electrical and functional subset of the MAB8048, there are some differences:

1. Pin out – As the MAB8021 is a 28-lead DIL, some form of adapter must be used to interface the MAB8021 socket to ICE-49. An emulation board (EM-1) has been designed to perform this function. The EM-1 also accounts for the increased flexibility of some MAB8021 I/O lines.
2. Instruction time – The MAB8021 instruction cycle is 30 clock cycles long, the MAB8048 instruction cycle is 15 clock cycles long. Where exact timing is important, the MAB8048 breadboard part should be operated at half the MAB8021 clock rate.
3. Test 1 – To facilitate developing time-of-day routine, and for thyristor control, the T1 pin (without the pull-up resistor option) will detect zero-crossing of a capacitively coupled a.c. input.
4. Quasi-bidirectional Ports – All MAB8021 ports are quasi-bidirectional to facilitate stand-alone use. Port 0 has open drain outputs and by mask option it may or may not have pull-up resistors.
5. Oscillator – The MAB8021 has an on-chip oscillator that is optimized for the single inductor mode. External connection will differ from the MAB8048.
6. Timer/counter – 1. If prescaler overflow occurs during a 'STRT T' or 'STOP TCNT' instruction, the MAB8048 will increment the timer, while the MAB8021 will not.
   2. The MAB8021 sets the timer flag in the same cycle as the overflow. The MAB8048 waits one cycle. Therefore, the MAB8021 can do a JTF instruction one cycle earlier (prescaler = '0') than the MAB8048 (prescaler = '1').
   3. The MAB8048 doesn't increment its timer in the second cycle of a 2-cycle instruction; the MAB8021 does.
7. High-current outputs – Very high current drive is desirable for minimizing external parts required to do high-power control. P10 and P11 have been designated high drive outputs capable of sinking 7 mA at V_{SS} + 2,5 V. (For clarity, this is a 7 mA to V_{SS} with a 2,5 V drop across the buffer.) These pins may, of course, be used in parallel for 14 mA drive if the output logic states are always the same.
8. Reset – Reset has been modified on the MAB8021. On the MAB8021, RESET is active HIGH; on the MAB8048, active LOW. Also, the MAB8021 does not reset the timer flag, while the MAB8048 does.
9. Instruction set – The following instructions (see Table 2), which are found in the MAB8048, have been deleted from the MAB8021 instruction set.

Table 2 Instruction set differences

DATA MOVES		REGISTERS		BRANCH		SUBROUTINE		CONTROL		INPUT/OUTPUT	
MOV	A,PSW	DEC	R	JT0	addr	RETR		EN	I	ANL	P,#data
MOV	PSW,A	FLAGS		JNT0	addr	TIMER		DIS	I	ORL	P,#data
MOVX	A,@R			JF0	addr			SEL	RB0	INS	A,BUS*
MOVX	@R,A	CLR	F0	JF1	addr	EN	TCNTI	SEL	RB1	OUTL	BUS,A*
MOVP3	A,@A	CPL	F0	JNI	addr	DIS	TCNTI	SEL	MB0	ANL	BUS,#data
•		CLR	F1	JBb	addr			SEL	MB1	ORL	BUS,#data
		CPL	F1					ENT0	CLK		

• These instructions have been replaced in the MAB8021 by INA,P0 and OUTL P0,A respectively. OUTL P0,A has op-code of MOVX @R0,A.

Table 3 Instruction set

	mnemonic		description	bytes	cycles
ACCUMULATOR	ADD	A,R	Add register to A	1	1
	ADD	A,@R	Add data memory to A	1	1
	ADD	A, # data	Add immediate to A	2	2
	ADDC	A,R	Add with carry	1	1
	ADDC	A,@R	Add with carry	1	1
	ADDC	A, # data	Add with carry	2	2
	ANL	A,R	AND register to A	1	1
	ANL	A,@R	AND data memory to A	1	1
	ANL	A, # data	AND immediate to A	2	2
	ORL	A,R	OR register to A	1	1
	ORL	A,@R	OR data memory to A	1	1
	ORL	A, # data	OR immediate to A	2	2
	XRL	A,R	Exclusive OR register to A	1	1
	XRL	A,@R	Exclusive OR data memory to A	1	1
	XRL	A, # data	Exclusive OR immediate to A	2	2
	INC	A	Increment A	1	1
	DEC	A	Decrement A	1	1
	CLR	A	Clear A	1	1
	CPL	A	Complement A	1	1
	DA	A	Decimal adjust A	1	1
	SWAP	A	Swap nibbles of A	1	1
	RL	A	Rotate A left	1	1
	RLC	A	Rotate A left through carry	1	1
	RR	A	Rotate A right	1	1
	RRC	A	Rotate A right through carry	1	1
INPUT/OUTPUT	IN	A,P	Input port to A	1	2
	OUTL	P,A	Output A to port	1	2
	MOVD	A,P	Input expander port to A	1	2
	MOVD	P,A	Output A to expander port	1	2
	ANLD	P,A	AND A to expander port	1	2
	ORLD	P,A	OR A to expander port	1	2

	mnemonic		description	bytes	cycles
REG.	INC	R	Increment register	1	1
	INC	@R	Increment data memory	1	1
BRANCH	JMP	addr	Jump unconditional	2	2
	JMPP	@A	Jump in page indirect	1	2
	DJNZ	R, addr	Decrement register and jump on R not zero	2	2
	JC	addr	Jump on carry = 1	2	2
	JNC	addr	Jump on carry = 0	2	2
	JZ	addr	Jump on A zero	2	2
	JNZ	addr	Jump on A not zero	2	2
	JT1	addr	Jump on T1 = 1	2	2
	JNT1	addr	Jump on T1 = 0	2	2
	JTF	addr	Jump on timer flag	2	2
SUBR.	CALL		Jump to subroutine	2	2
	RET		Return	1	2
FLAGS	CLR	C	Clear Carry	1	1
	CPL	C	Complement Carry	1	1
DATA MOVES	MOV	A,R	Move register to A	1	1
	MOV	A,@R	Move data memory to A	1	1
	MOV	A, # data	Move immediate to A	2	2
	MOV	R,A	Move A to register	1	1
	MOV	@R,A	Move A to data memory	1	1
	MOV	R, # data	Move immediate to register	2	2
	MOV	@R, # data	Move immediate to data memory	2	2
	XCH	A,R	Exchange A and register	1	1
	XCH	A, @R	Exchange A and data memory	1	1
	XCHD	A,@R	Exchange lower nibble of A and data memory	1	1
	MOVP	A,@A	Move to A from current page	1	2
TIMER/ COUNTER	MOV	A,T	Read timer/counter	1	1
	MOV	T,A	Load timer/counter	1	1
	STRT	T	Start timer	1	1
	STRT	CNT	Start counter	1	1
	STOP	TCNT	Stop timer/counter	1	1
	NOP		No operation	1	1

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage with respect to V_{SS}	V_{CC}	−0,5 to + 7	V
Voltage on any input or output	V_I, V_O	−0,5 to + 7	V
D.C. current into any input or output	± I_I, ± I_O	max. 10	mA
Total power dissipation	P_{tot}	max. 1	W
Storage temperature range	T_{stg}	−65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

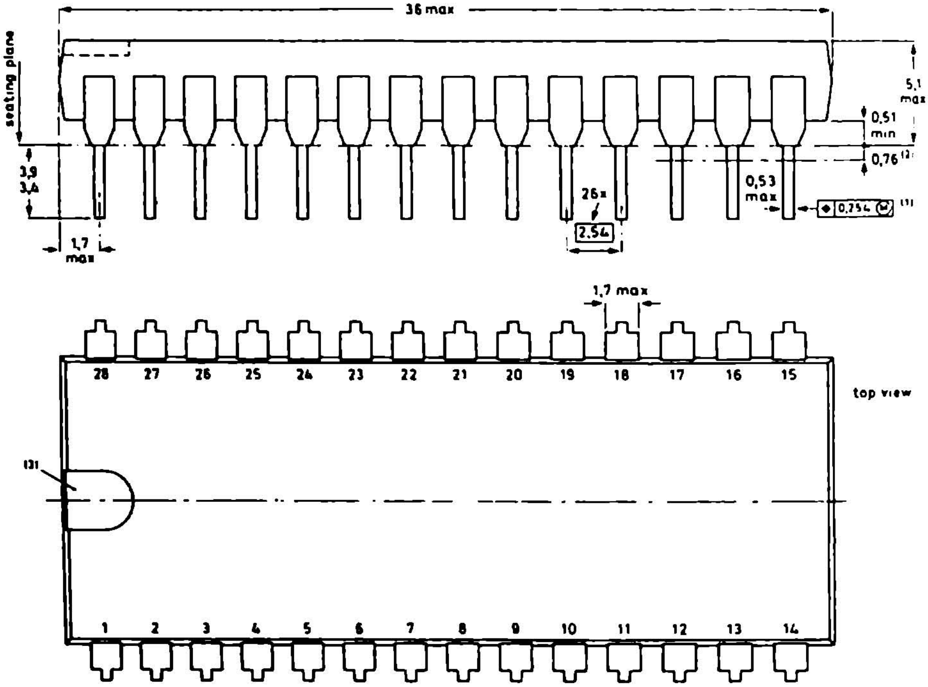
	symbol	min.	typ.	max.		conditions
Supply voltage	V_{CC}	4,5	5,5	6,5	V	
Supply current	I_{CC}	–	–	75	mA	
Input voltage LOW	V_{IL}	−0,5	–	0,8	V	
Input voltage HIGH all inputs except XTAL1,XTAL2,T1,RESET	V_{IH}	3,0	–	V_{CC}	V	V_{CC} = 5,5 + 1 V
Input voltage HIGH XTAL1,XTAL2,T1,RESET	V_{IH}	3,8	–	V_{CC}	V	V_{CC} = 5,5 + 1 V
Input voltage HIGH (10%) all inputs except XTAL1,XTAL2,T1,RESET	V_{IH}	2,0	–	V_{CC}	V	V_{CC} = 5,0 V ± 10%
Input voltage HIGH (10%) XTAL1,XTAL2,T1,RESET	V_{IH}	3,5	–	V_{CC}	V	V_{CC} = 5,0 V ± 10%
Output voltage LOW	V_{OL}	–	–	0,45	V	I_{OL} = 1,6 mA
Output voltage LOW P10, P11	V_{OL}	–	–	2,5	V	I_{OL} = 7 mA
Output voltage HIGH all outputs unless open drain	V_{OH}	2,4	–	–	V	$-I_{OH}$ = 50 μA
Output leakage current open drain option-port 0	± I_{OL}	–	–	10	μA	$V_{CC} > V_I > V_{SS} + 0{,}45$ V

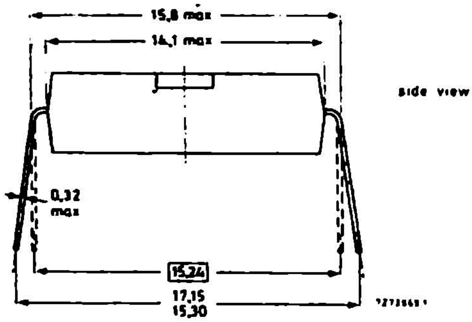
A.C. CHARACTERISTICS

V_{SS} = 0 V; V_{CC} = 5,5 V ± 1 V; T_{amb} = 0 to + 70 °C; unless otherwise specified

	symbol	min.	typ.	max.		
Cycle time	t_{cy}	8,38	–	50,0	μs	3,58 MHz crystal = 8,38 μs
T1 zero-cross characteristics						
Zero-cross detection input (T1) voltage; peak-to-peak value	$V_{zx(p-p)}$	....			V	a.c. coupled; C = 0,2 μF
Zero-cross accuracy 50 Hz sine-wave	A_{zx}	....	....	....	mV	
Zero-cross detection input frequency (T1)	f_{T1}	....	....	....	kHz	

28-LEAD DUAL IN-LINE; PLASTIC (SOT-117)


36 max
seating plane
5,1 max
0,51 min
0,76 (2)
3,9
3,4
0,53 max
26x
2,54
0,254 M (1)
1,7 max
1,7 max
28 27 26 25 24 23 22 21 20 19 18 17 16 15
top view
(3)
1 2 3 4 5 6 7 8 9 10 11 12 13 14


15,8 max
14,1 max
side view
0,32 max
15,24
17,15
15,30

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

Dimensions in mm

SOLDERING

See next page.

SOLDERING

1. By hand

Apply the soldering iron below the seating plane (or not more than 2 mm above it).
If its temperature is below 300 °C it must not be in contact for more than 10 seconds; if between 300 °C and 400 °C, for not more than 5 seconds.

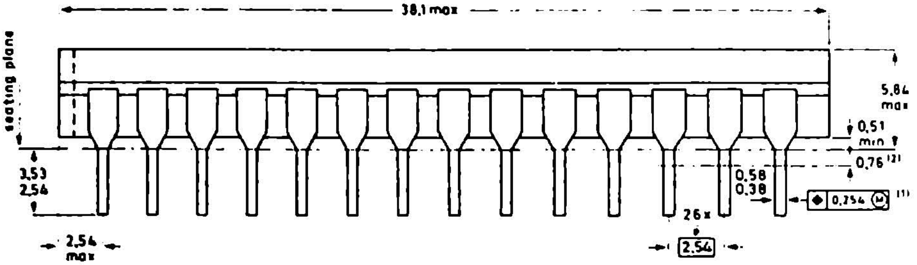
2. By dip or wave

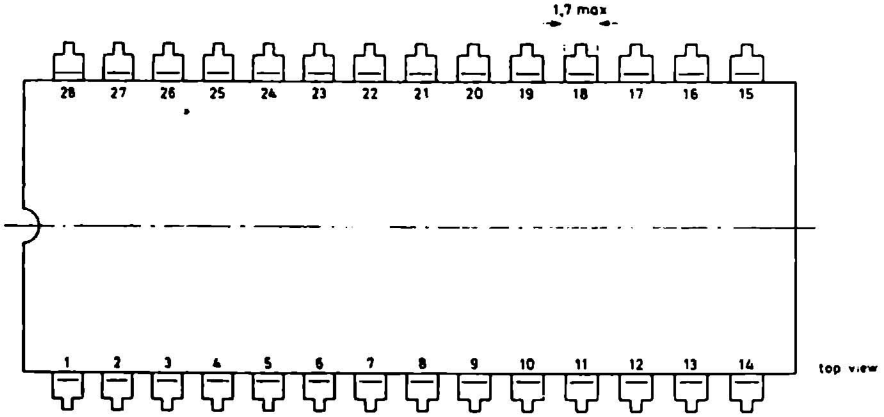
The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.
The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

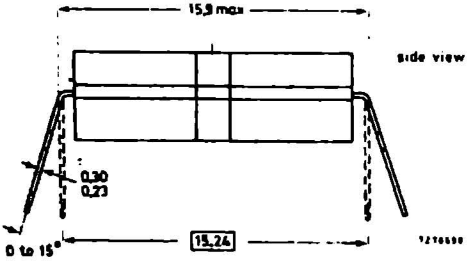
3. Repairing soldered joints

The same precautions and limits apply as in (1) above.

28-LEAD DUAL IN-LINE; CERAMIC (SOT-135)


38,1 max
seating plane
5,84 max
0,51 min
0,76 (2)
0,58
0,38
0,254 M (1)
26 x
2,54
3,53
2,54
2,54 max


1,7 max
28 27 26 25 24 23 22 21 20 19 18 17 16 15
1 2 3 4 5 6 7 8 9 10 11 12 13 14
top view


15,9 max
side view
0,30
0,23
0 to 15°
15,24

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

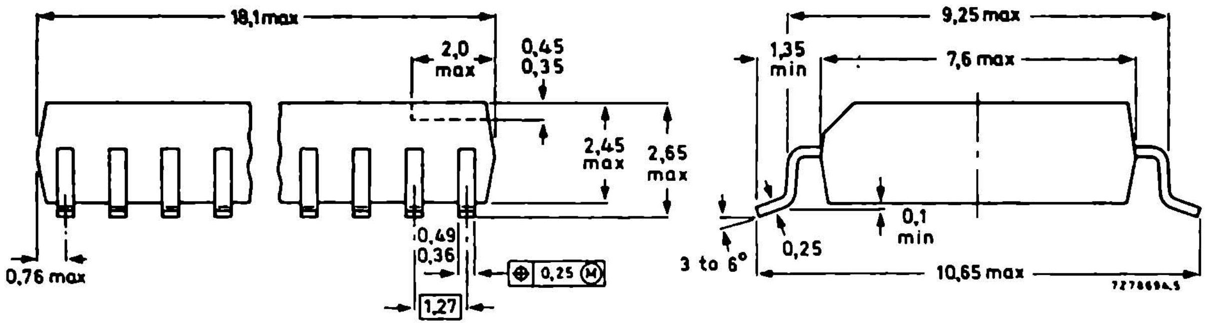
(2) Lead spacing tolerances apply from seating plane to the line indicated.

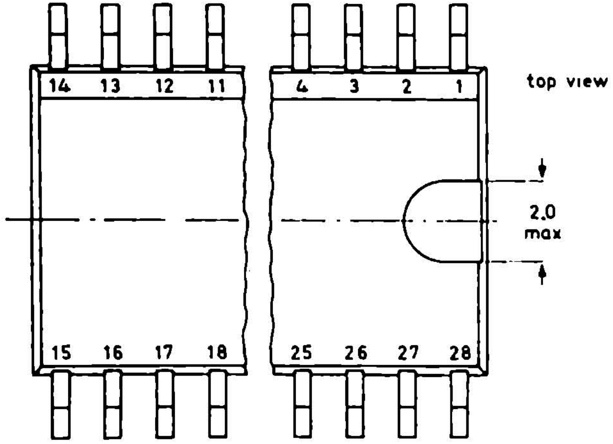
Dimensions in mm

Remarks

1. Leads are given positive misalignment so that they grip after insertion.
2. Leads are Ni-Fe, pure tin plated.

28-LEAD FLAT PACK; PLASTIC (SO-28; SOT-136A)


18,1max
2,0
max
0,45
0,35
2,45
max
2,65
max
0,76 max
0,49
0,36
1,27
⊕ 0,25 Ⓜ
9,25 max
1,35
min
7,6 max
0,1
min
0,25
3 to 6°
10,65 max
7278694.5


top view
14
13
12
11
4
3
2
1
2,0
max
15
16
17
18
25
26
27
28

Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

SOLDERING

See next page.

SOLDERING

The reflow solder technique

The preferred technique for mounting miniature components on hybrid thick or thin-film circuits is reflow soldering. Solder is applied to the required areas on the substrate by dipping in a solder bath or, more usually, by screen printing a solder paste. Components are put in place and the solder is reflowed by heating.

Solder pastes consist of very finely powdered solder and flux suspended in an organic liquid binder. They are available in various forms depending on the specification of the solder and the type of binder used. For hybrid circuit use, a tin-lead solder with 2 to 4% silver is recommended. The working temperature of this paste is about 220 to 230 °C when a mild flux is used.

For printing the paste onto the substrate a stainless steel screen with a mesh of 80 to 105 µm is used for which the emulsion thickness should be about 50 µm. To ensure that sufficient solder paste is applied to the substrate, the screen aperture should be slightly larger than the corresponding contact area.

The contact pins are positioned on the substrate, the slight adhesive force of the solder paste being sufficient to keep them in place. The substrate is heated to the solder working temperature preferably by means of a controlled hot plate. The soldering process should be kept as short as possible: 10 to 15 seconds is sufficient to ensure good solder joints and evaporation of the binder fluid.
After soldering, the substrate must be cleaned of any remaining flux.

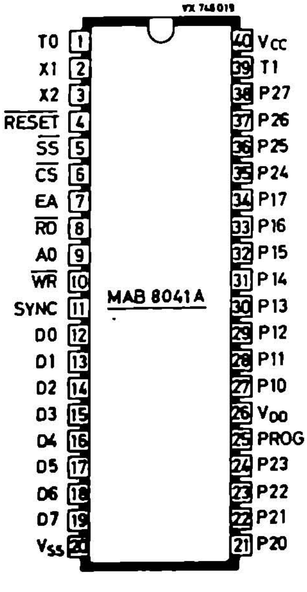
SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The **MAB 8041 A** is an 8-bit microcomputer with an architecture based upon the 8-bit microcomputer MAB 8048. It contains a low-cost microcomputer, program memory, I/O ports, timer/counter and clock oscillator in a single 40-pin package. Special interface registers are included which enable the UPI to function as a peripheral device to an 8-bit master processor.

FEATURES

- 8-bit CPU
- 1 K x 8 ROM (program memory)
- 64 x 8 RAM (data memory)
- 18 programmable I/O pins
- 8-bit interval timer / event counter
- one 8-bit status- and two data-registers for asynchronous slave-to-master interface
- DMA, interrupt or polled-operation support
- RAM power-down capability


7Z 74019
T0 1
X1 2
X2 3
RESET 4
SS 5
CS 6
EA 7
RD 8
A0 9
WR 10
SYNC 11
D0 12
D1 13
D2 14
D3 15
D4 16
D5 17
D6 18
D7 19
VSS 20
40 VCC
39 T1
38 P27
37 P26
36 P25
35 P24
34 P17
33 P16
32 P15
31 P14
30 P13
29 P12
28 P11
27 P10
26 VDD
25 PROG
24 P23
23 P22
22 P21
21 P20
MAB 8041 A

QUICK REFERENCE DATA

Supply voltage

$V_{DD} = 5\ V$

Total supply current

$I_{DD} + I_{CC} = 60\ mA$

Program memory (internally)

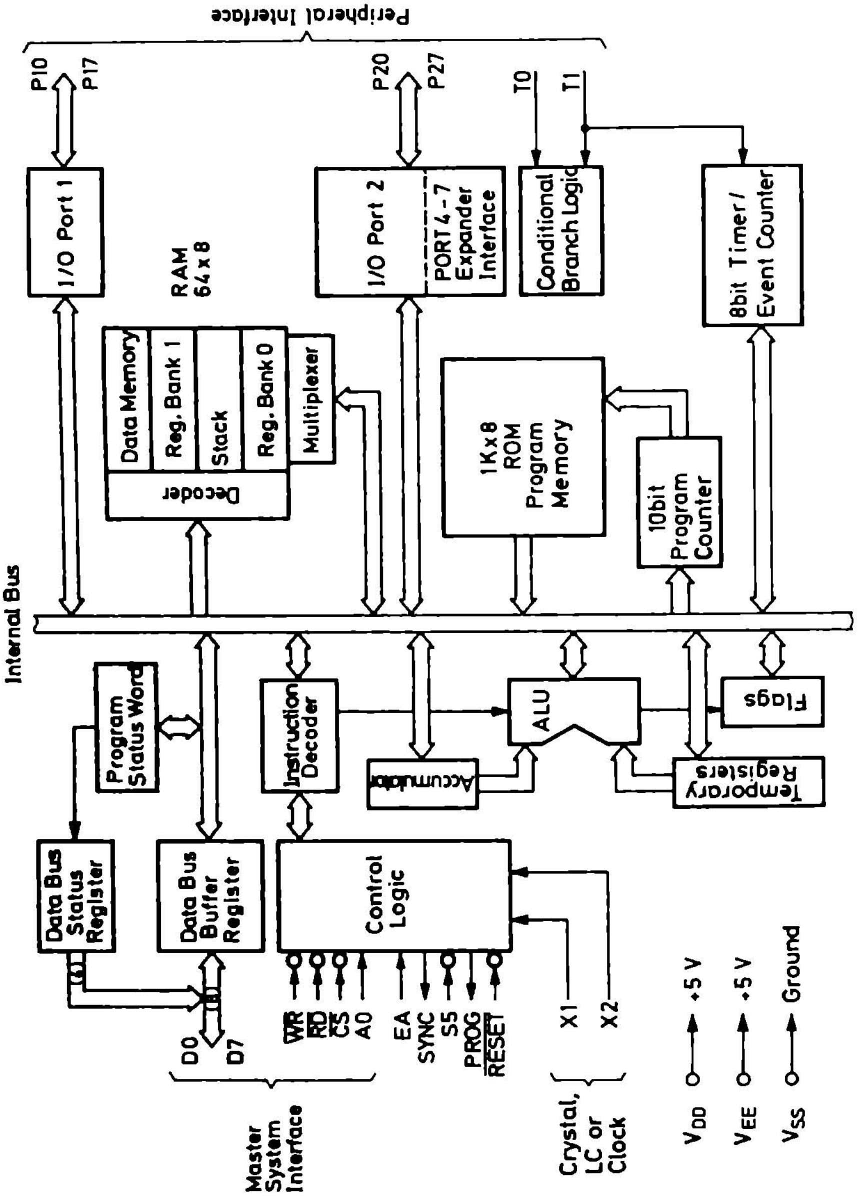
1024 x 8 bit

Data memory (internally)

64 x 8 bit

Cycle time (at 6 MHz)

$t = 2,5\ \mu s$


Peripheral Interface
P10
P17
P20
P27
T0
T1
I/O Port 1
I/O Port 2
PORT 4-7 Expander Interface
Conditional Branch Logic
8bit Timer / Event Counter
RAM 64 x 8
Data Memory
Reg. Bank 1
Stack
Reg. Bank 0
Multiplexer
Decoder
1K x 8 ROM Program Memory
10bit Program Counter
Internal Bus
Program Status Word
Instruction Decoder
ALU
Flags
Accumulator
Temporary Registers
Data Bus Status Register
Data Bus Buffer Register
Control Logic
D0
D7
WR
RD
CS
A0
EA
SYNC
SS
PROG
RESET
X1
X2
Master System Interface
Crystal, LC or Clock
V_{DD}
+5 V
V_{EE}
+5 V
V_{SS}
Ground

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The MAB8048H family of single-chip 8-bit microcomputers are fabricated in H-MOS.
Two interchangeable (pin compatible) versions are available:

- The **MAB8048H** with resident mask-programmed ROM,
- The **MAB8035HL** without resident program memory for use with external EPROM/ROM.

The MAB8048H family are designed to be efficient control processors as well as arithmetic processors. Their instruction set allows the user to directly set and reset individual I/O lines as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions enable efficient implementation of standard logic functions. Code efficiency is high; over 70% of the instructions are single byte; all others are two byte.

An on-chip 8-bit counter is provided, which can count either machine cycles (÷ 32) or external events. The counter can be programmed to cause an interrupt to the processor.
Program and data memories can be expanded using standard devices. Input/output capabilities can be expanded using standard devices.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 40-pin package
- 1 K x 8 ROM, 64 x 8 RAM, 27 I/O lines
- Internal counter/timer
- Internal oscillator, clock driver
- Single-level interrupts: external and counter/timer
- 17 internal registers: accumulator, 16 addressable registers
- Over 90 instructions: 70% single byte
- All instructions: 1 or 2 cycles (1,875 µs per cycle)
- Easily expandable memory and I/O
- TTL compatible inputs and outputs
- Single 5 V supply

APPLICATIONS

- Peripheral interfaces and controllers
- Test and measurement instruments
- Sequencers
- Audio/video systems
- Environmental control systems
- Modems and data enciphering

PACKAGE OUTLINES

MAB 8048 HP: 40-lead DIL; plastic (SOT-129).
MAB 8048 HD: 40-lead DIL; ceramic (SOT-145).
MAB 8048 HD: 40-lead DIL; metal-ceramic (SOT-88B).
MAB 8048 HT: 40-lead flat-pack; plastic (VSO-40, SOT-158).

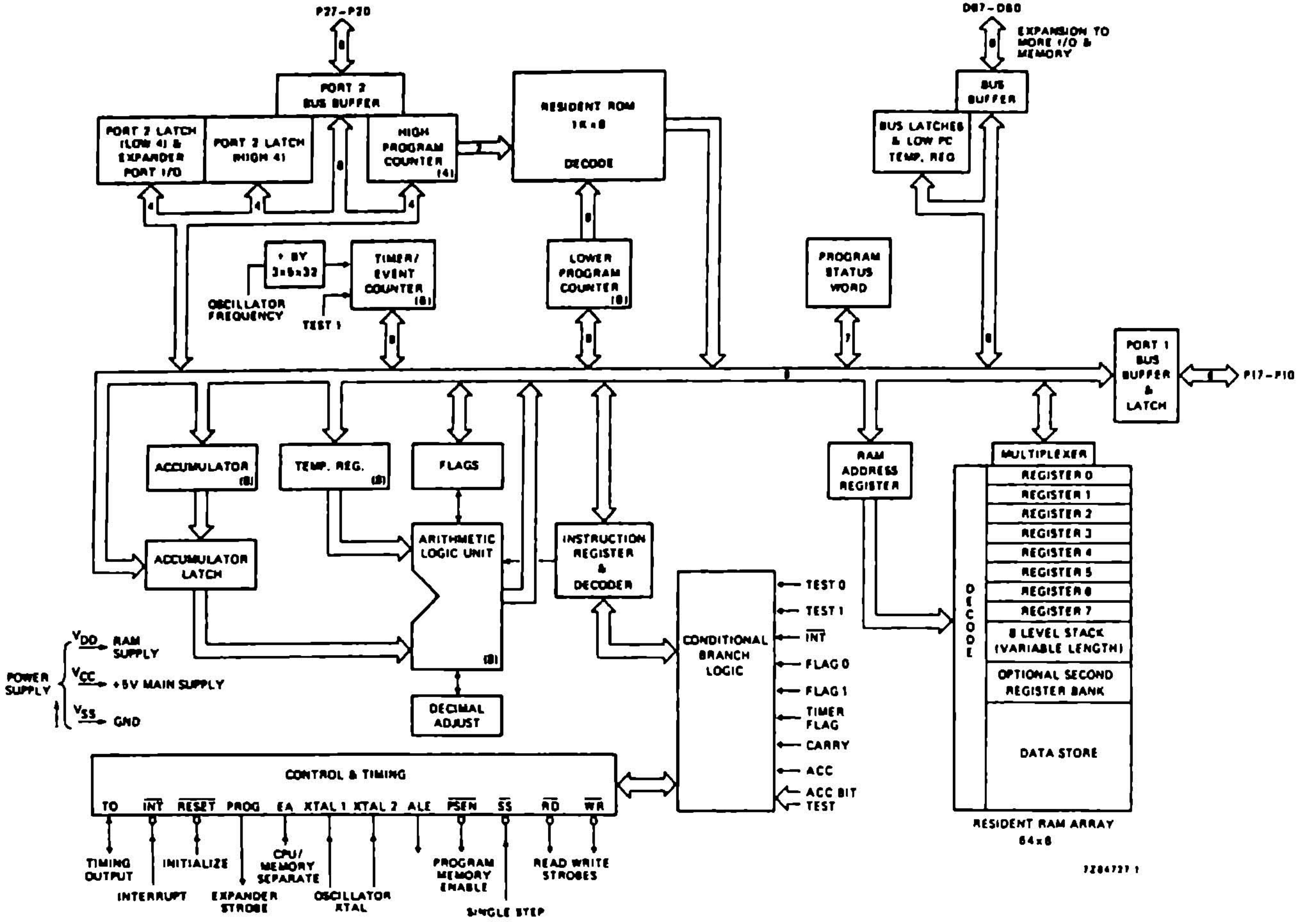
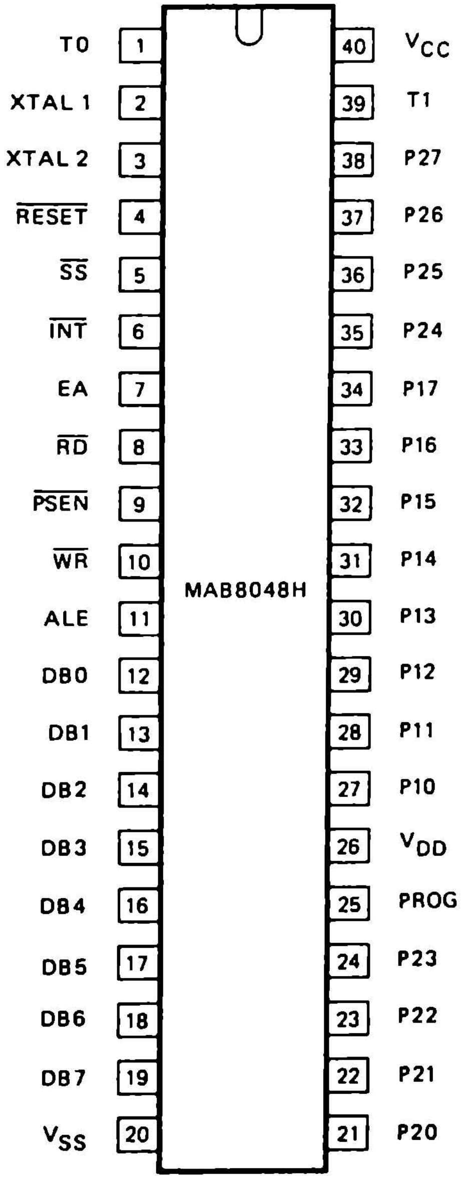

P27–P20
8
PORT 2 BUS BUFFER
PORT 2 LATCH (LOW 4) & EXPANDER PORT I/O
PORT 2 LATCH (HIGH 4)
HIGH PROGRAM COUNTER (4)
4
4
8
4
RESIDENT ROM 1K x 8
DECODE
8
÷ BY 3x5x32
OSCILLATOR FREQUENCY
TIMER/ EVENT COUNTER (8)
TEST 1
8
LOWER PROGRAM COUNTER (8)
8
PROGRAM STATUS WORD
7
DB7–DB0
8
EXPANSION TO MORE I/O & MEMORY
BUS BUFFER
BUS LATCHES & LOW PC TEMP. REG
8
8
PORT 1 BUS BUFFER & LATCH
8
P17–P10
ACCUMULATOR (8)
TEMP. REG. (8)
FLAGS
ACCUMULATOR LATCH
ARITHMETIC LOGIC UNIT (8)
DECIMAL ADJUST
INSTRUCTION REGISTER & DECODER
RAM ADDRESS REGISTER
MULTIPLEXER
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
8 LEVEL STACK (VARIABLE LENGTH)
OPTIONAL SECOND REGISTER BANK
DATA STORE
DECODE
RESIDENT RAM ARRAY 64 x 8
CONDITIONAL BRANCH LOGIC
TEST 0
TEST 1
INT
FLAG 0
FLAG 1
TIMER FLAG
CARRY
ACC
ACC BIT TEST
POWER SUPPLY
V_{DD} RAM SUPPLY
V_{CC} +5V MAIN SUPPLY
V_{SS} GND
CONTROL & TIMING
TO
$\overline{INT}$
$\overline{RESET}$
PROG
EA
XTAL 1
XTAL 2
ALE
$\overline{PSEN}$
$\overline{SS}$
$\overline{RD}$
$\overline{WR}$
TIMING OUTPUT
INTERRUPT
INITIALIZE
EXPANDER STROBE
CPU/ MEMORY SEPARATE
OSCILLATOR XTAL
PROGRAM MEMORY ENABLE
SINGLE STEP
READ WRITE STROBES
7Z84727.1

Fig. 1 Block diagram.


T0
XTAL 1
XTAL 2
$\overline{RESET}$
$\overline{SS}$
$\overline{INT}$
EA
$\overline{RD}$
$\overline{PSEN}$
$\overline{WR}$
ALE
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
V_{SS}
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
MAB8048H
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
V_{CC}
T1
P27
P26
P25
P24
P17
P16
P15
P14
P13
P12
P11
P10
V_{DD}
PROG
P23
P22
P21
P20

7Z84742.1

Fig. 2 Pinning diagram; for pin designation see next page.

PIN DESIGNATION

designation	pin no.	function
DB0–DB7	12–19	**BUS.** Bidirectional I/O port that can be read or written using the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes. This port can also be statically latched. Contains the 8 lower order address bits during external memory access and receives the addressed instruction under control of $\overline{\text{PSEN}}$. $\overline{\text{PSEN}}$, ALE, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ determine whether the access is an instruction fetch or a read/write access to external RAM.
P10–P17	27–34	**Port 1.** 8-bit quasi-bidirectional I/O port (note 1)
P20–P27	21–24, 35–38	**Port 2.** 8-bit quasi-bidirectional I/O port (note 1). P20–P23 contain the 4 higher order address bits during an access of external program memory and also serve as a 4-bit I/O expander bus for the 8243.
PROG	25	**Output strobe** (active LOW) for 8243 I/O expander.
T0	1	**Input pin** sensed using the JT0 and JNT0 instructions. **Clock output pin** when designated as such by the ENT0 CLK instruction.
T1	39	**Input pin** sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
$\overline{\text{INT}}$	6	**Interrupt input pin.** When LOW causes an interrupt in the current program if external interrupt is enabled. Can also be used as an input, testable using the JNI instruction. Interrupt is disabled during and after a RESET.
$\overline{\text{RESET}}$	4	**Reset input** pin used to initialize the microcomputer. Active LOW. During program verification the address is latched by a '0' to '1' transition on $\overline{\text{RESET}}$ and the data at the addressed location is output on BUS (note 2).
ALE	11	**Address latch enable.** Occurs each clock cycle and is useful for timing and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0 to DB7 outputs.
$\overline{\text{RD}}$	8	**Read BUS.** Active LOW strobe used to gate data onto BUS lines when reading from an external source.
$\overline{\text{WR}}$	10	**Write BUS.** Active LOW strobe used to write data from BUS lines to an external designation.
EA	7	**External access input.** When HIGH, forces instruction fetch from external memory.
$\overline{\text{PSEN}}$	9	**Program store enable.** Active LOW strobe that occurs only during a fetch from external program memory.
$\overline{\text{SS}}$	5	**Single step input.** Active LOW which is used with ALE to cause the microcomputer to execute a single instruction.
XTAL1	2	**One side of crystal** (or inductor) input for internal oscillator. Can also be used as an input for an external timing source (note 2).

designation	pin no.	function
XTAL2	3	Other side of crystal.
V_{SS}	20	Ground.
V_{CC}	40	Power supply, + 5 V.
V_{DD}	26	RAM standby power supply, low power pin.

Notes

1. Each port line can be designated as an input or an output. A line is designated as an input by first writing a logic '1' to the line. $\overline{\text{RESET}}$ sets all lines to logic '1'.
2. Non-standard TTL V_{IH}.

FUNCTIONAL DESCRIPTION

Detailed information available on request.

Program memory

The resident program memory consists of a 1024 byte ROM (MAB8048H); the MAB8035HL has no resident program memory.
The total addressing capability is 4096 bytes.
The program memory address space is divided into two 2048-bytes banks MB0 and MB1 (Fig. 3).
Further, the program memory is divided into pages of 256 bytes each. This latter division applies only for conditional branches.
There are three locations in program memory of special importance. These locations contain the first instruction to be executed after one of three events.
The three locations and their contents are:

location 0 — activation, then deactivation of the $\overline{\text{RESET}}$ line.
location 3 — activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled,
location 7 — an overflow of the timer/counter if the T/C interrupt is enabled.

Data memory

The resident data memory, as shown in Fig. 4, consists of a 64 byte RAM. All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1. The first eight locations of RAM (0 to 7) are designated as working registers and are directly addressable by several instructions. By selecting register bank 1, RAM locations 24 to 31 become the working registers, replacing those in register bank 0 (0 to 7).
RAM locations 8 to 23 are designated as the stack. Two locations (bytes) are used per CALL, allowing up to eight levels of subroutine nesting.
If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM (greater than 320 bytes total) is required, an I/O port can be used to select one (256 byte) bank of external memory at a time.

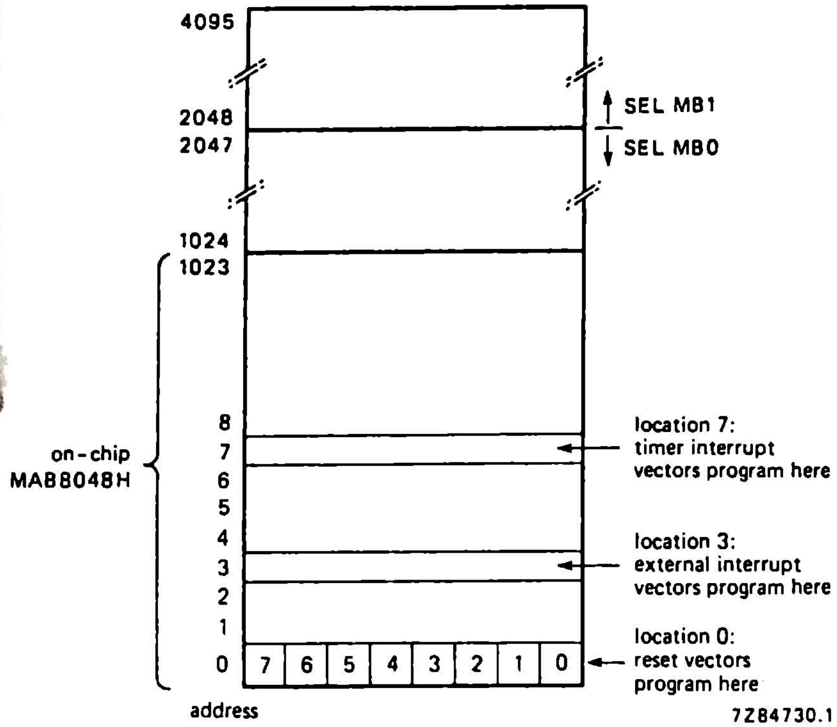

4095
2048
2047
1024
1023
8
7
6
5
4
3
2
1
0
address
7
6
5
4
3
2
1
0
SEL MB1
SEL MB0
on-chip
MAB8048H
location 7:
timer interrupt
vectors program here
location 3:
external interrupt
vectors program here
location 0:
reset vectors
program here
7Z84730.1

Fig. 3 Program memory map.

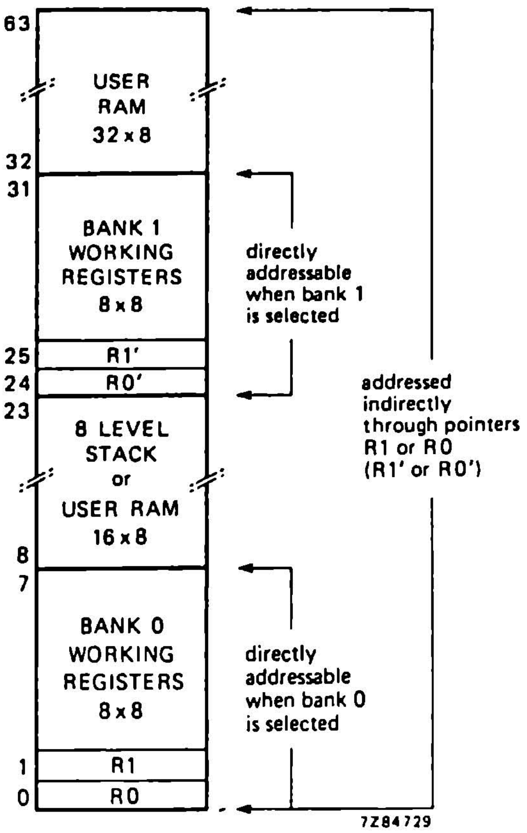

63
32
31
25
24
23
8
7
1
0
USER
RAM
32 x 8
BANK 1
WORKING
REGISTERS
8 x 8
R1'
R0'
8 LEVEL
STACK
or
USER RAM
16 x 8
BANK 0
WORKING
REGISTERS
8 x 8
R1
R0
directly
addressable
when bank 1
is selected
addressed
indirectly
through pointers
R1 or R0
(R1' or R0')
directly
addressable
when bank 0
is selected
7Z84729

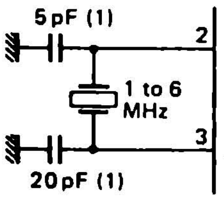
Fig. 4 Data memory map. In addition R0 or R1 (R0' or R1') may be used to address 256 words of an external RAM.

Program counter and stack

The program counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. When EA is '0', the PC can address locations 0 to 1023 of internal program memory. At the 1 K boundary, an automatic switch-over to external memory is made. When EA is '1', all the program is fetched from external ROM/EPROM. The total address space is 4 K bytes.
An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the program status word (PSW). Data RAM locations 8 to 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000B, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.
The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Oscillator and clock

The MAB8048H contains its own internal oscillator and clock driver. A crystal, inductor or external pulse generator determines the oscillator frequency (see Figs 5, 6 and 7). The output of the oscillator is divided-by-three and is available at T0 (pin 1) by executing the ENT0 CLK instruction. This CLK signal is divided-by-five to define a machine (instruction) cycle. It is available at ALE (pin 11).



(1) Including crystal-socket stray capacitance.

Fig. 5 Crystal oscillator mode. Crystal series impedance should be < 75 Ω at 6 MHz and < 180 Ω at 3,6 MHz.

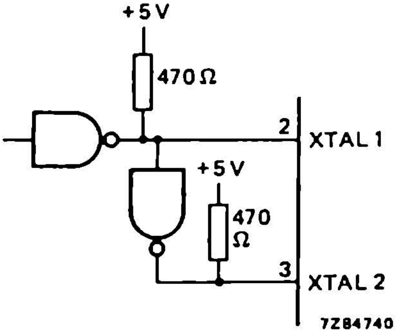
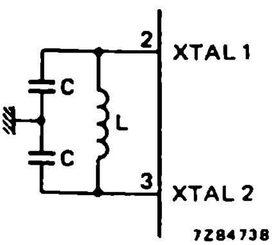


Fig. 6 Driving from external source. Both XTAL1 and XTAL2 should be driven. Resistors to V_{CC} (+ 5 V) are needed to ensure $V_{IH} = 3{,}8$ V if TTL circuitry is used. The minimum HIGH and LOW times are 45%.



L (µH)	C (pF)	nom. f (MHz)
45	20	5,2
120	20	3,2

$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3C_{pp}}{2}$$

Fig. 7 LC oscillator mode. Each C should be ≈ 20 pF including stray capacitance.
$C_{pp} \approx 5$ to 10 pF (pin-to-pin capacitance).

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided-by-32 before they are applied to the input of the 8-bit counter. External events are applied directly to the input of the counter. The maximum frequency that can be counted is one third of the machine cycle frequency. The minimum positive duty cycle that can be detected is 0,2 times the cycle period. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

Interrupt

An interrupt may be generated by either an external input ($\overline{\mathrm{INT}}$, pin 6) or the overflow of the internal counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine. After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

Input/output

The MAB8048H has 27 I/O lines. These lines are arranged as three 8-line ports, which serve as either inputs, outputs or bidirectional ports and 3 'test' inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, e.g., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The circuit configuration is shown in Fig. 8. Each line is continuously pulled up to + 5 V through a relative high resistance (≈ 50 kΩ). This pull-up is sufficient to provide the source current for a TTL HIGH level, yet can be pulled LOW by a standard TTL gate, thus allowing the same pin to be used for both input and output. To provide fast switching times during '0' to '1' transition a relatively low impedance transistor (≈ 5 kΩ) is switched on momentarily for 1/5 of a machine cycle whenever a '1' is written to the line. When a '0' is written, a low impedance (≈ 300 Ω) transistor overcomes the light pull-up and provides TTL current sinking capability.

Since the pull-down transistor is a low-impedance device, a '1' must first be written to any line which is to be used as an input. RESET initializes all lines to the high impedance '1' state. This structure allows input and output on the same pin and also allows a mixture of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

BUS

BUS is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.

As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{\mathrm{RD}}$ and $\overline{\mathrm{WR}}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{\mathrm{WR}}$ output line and output data is valid at the trailing-edge of $\overline{\mathrm{WR}}$. A read of the port generates a pulse on the $\overline{\mathrm{RD}}$ output line and input data must be valid at the trailing-edge of $\overline{\mathrm{RD}}$. When not being written or read, the BUS lines are high impedance.

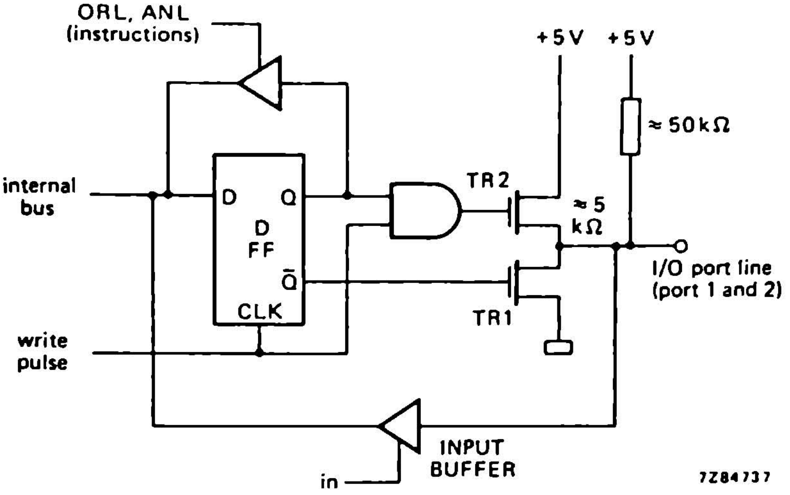


Fig. 8 Quasi-bidirectional port structure.

Test (T0, T1) and $\overline{\text{INT}}$ inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These pins are T0, T1, and $\overline{\text{INT}}$ and they allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1 and $\overline{\text{INT}}$ pins have other possible functions as well.

$\overline{\text{RESET}}$ input

The $\overline{\text{RESET}}$ input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pull-up resistor which, in combination with an external 1 μF capacitor, provides an internal reset pulse of sufficient duration to guarantee all circuitry is reset. If the reset pulse is generated externally, the $\overline{\text{RESET}}$ pin must be held at ground (0,45 V) for at least 10 ms after the power supply is within tolerance. Only five machine cycles (12,5 μs at 6 MHz) are required if power is already on and the oscillator has stabilized. Typical circuitry is shown in Fig. 9.

Single step input ($\overline{\text{SS}}$)

By proper control of the $\overline{\text{SS}}$ line, the processor can be forced to execute one instruction and then to wait until the single step switch is activated again.

Power-down mode

In the MAB8048H and MAB8035HL, power can be removed from all but the 64 x 8 data RAM array, for low power standby operation. In the power-down mode the contents of the data RAM can be maintained while drawing typically 10% to 15% of the normal operating power supply voltage.
V_{CC} serves as the + 5 V supply pin for the bulk of the circuitry, while the V_{DD} pin supplies only the RAM array. In normal operation, both pins are at + 5 V. In standby, V_{CC} is at ground and only V_{DD} is maintained at + 5 V.
Applying $\overline{\text{RESET}}$ to the processor through the $\overline{\text{RESET}}$ pin inhibits any access to the RAM by the processor and guarantees that the RAM cannot be inadvertently altered as power is removed from V_{CC}. A typical power down sequence occurs as shown in Fig. 10.

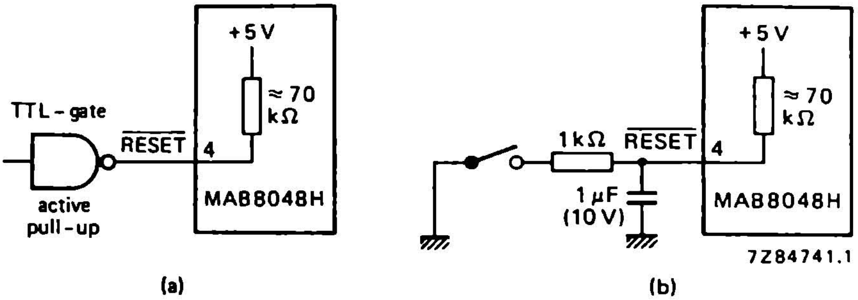

+5V
≈ 70 kΩ
TTL-gate
RESET
4
MAB8048H
active pull-up
(a)
+5V
≈ 70 kΩ
1 kΩ
RESET
4
1 μF (10 V)
MAB8048H
7Z84741.1
(b)

Fig. 9 An external reset circuit is shown in (a) and power-on reset in (b).

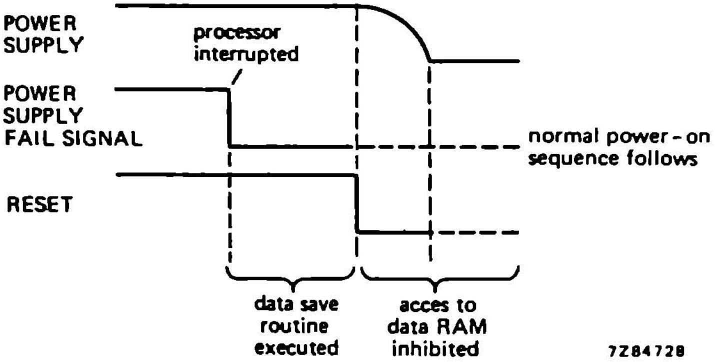

POWER SUPPLY
processor interrupted
POWER SUPPLY FAIL SIGNAL
normal power-on sequence follows
RESET
data save routine executed
acces to data RAM inhibited
7Z84728

Fig. 10 Power down sequence.

Instruction set

The MAB8048H instruction set consists of over 90 one and two-byte instructions (see Table 1). Program code efficiency is high because:

- working registers and program variables are stored in the RAM locations 0 to 63, which require only a single byte to address,
- program memory is divided into pages of 256 bytes, which means that branch destination addresses require one byte.

The instruction set efficiently manipulates and tests bits in addition to performing logical and arithmetic operations upon and the testing of bytes. A set of MOVE instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi-way branch (up to 256) upon the content of the accumulator to addresses stored in a look-up table. The 'decrement register and jump if not zero' instruction saves a byte every time it is used as opposed to using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The MAB8048H can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are highly desirable for real-time applications. See Table 2 for instruction timing.

Table 1 Instruction set is shown on the next 7 pages.

Symbols definitions used in Table 1.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number or expression (8-bits)
F0, F1	flags 0 and 1
I	interrupt
$\overline{\text{INT}}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1,	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
$	current value of program counter
←	is replaced by
↔	is exchanged with

Notes to Table 1.

1. Instruction code designations r and p form the binary representation of the registers and ports involved.
2. The dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

	mnemonic	function	description	Instruction code D7	D6	D5	D4	D3	D2	D1	D0	cycles	bytes	flags C	AC	F0	F1	BS
ACCUMULATOR	ADD A,Rr	(A)←(A) + (Rr) for r = 0–7	Add contents of designated register to A	0	1	1	0	1	r	r	r	1	1	●	●			
	ADD A,@Rr	(A)←(A) + ((Rr)) for r = 0–1	Add indirect the contents of the data memory location to A	0	1	1	0	0	0	0	r	1	1	●	●			
	ADD A,#data	(A)←(A) + data	Add immediate data to A	0 d7	0 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	2	2	●	●			
	ADDC A,Rr	(A)←(A) + (C) + (Rr) for r = 0–7	Add with carry the contents of designated register to A	0	1	1	1	1	r	r	r	1	1	●	●			
	ADDC A,@Rr	(A)←(A) + (C) + ((Rr)) for r = 0–1	Add indirect with carry the contents of the data memory location to A	0	1	1	1	0	0	0	r	1	1	●	●			
	ADDC A,#data	(A)←(A) + (C) + data	Add immediate data with carry to A	0 d7	0 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	2	2	●	●			
	ANL A,Rr	(A)←(A) AND (Rr) for r = 0–7	Logical AND contents of designated register with A	0	1	0	1	1	r	r	r	1	1					
	ANL A,@Rr	(A)←(A) AND ((Rr)) for r = 0–1	Logical AND indirect the contents of data memory with A	0	1	0	1	0	0	0	r	1	1					
	ANL A,#data	(A)←(A) AND data	Logical AND immediate data with A	0 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	2	2					
	CLR A	(A)←0	Clear the contents of A	0	0	1	0	0	1	1	1	1	1					
	CPL A	(A)←NOT(A)	Complement the contents of A	0	0	1	1	0	1	1	1	1	1					
	DA A		Decimal adjust the contents of A	0	1	0	1	0	1	1	1	1	1	●	●			
	DEC A	(A)←(A) −1	Decrement the contents of A by 1	0	0	0	0	0	1	1	1	1	1					
	INC A	(A)←(A) + 1	Increment the contents of A by 1	0	0	0	1	0	1	1	1	1	1					
	ORL A,Rr	(A)←(A) OR (Rr) for r = 0–7	Logical OR contents of designated register with A	0	1	0	0	1	r	r	r	1	1					
	ORL A,@Rr	(A)←(A) OR ((Rr)) for r = 0–1	Logical OR indirect the contents of data memory location with A	0	1	0	0	0	0	0	r	1	1					

ACCUMULATOR (continued)	ORL A,#data	(A)←(A) OR data	Logical OR immediate data with A	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2						
	RL A	(A_{n+1})←(A_n) (A_0)←(A_7) for n = 0–6	Rotate A left by 1-bit without carry	1	1	1	0	0	1	1	1	1	1						
	RLC A	(A_{n+1})←(A_n) (A_0)←(C) (C)←(A_7) for n = 0–6	Rotate A left by 1-bit through carry	1	1	1	1	0	1	1	1	1	1	●					
	RR A	(A_n)←(A_{n+1}) (A_7)←(A_0) for n = 0–6	Rotate A right by 1-bit without carry	0	1	1	1	0	1	1	1	1	1						
	RRC A	(A_n)←(A_{n+1}) (A_7)←(C) (C)←(A_0) n = 0–6	Rotate A right by 1-bit through carry	0	1	1	0	0	1	1	1	1	1	●					
	SWAP A	(A_{4-7})↔(A_{0-3})	Swap the two 4-bit nibbles in A	0	1	0	0	0	1	1	1	1	1						
	XRL A,Rr	(A)←(A) XOR (Rr)	Logical XOR contents of designated register with A	1	1	0	1	1	r	r	r	1	1						
	XRL A,@Rr	(A)←(A) XOR ((Rr))	Logical XOR indirect the contents of data memory location with A	1	1	0	1	0	0	0	r	1	1						
	XRL A,#data	(A)←(A) XOR data	Logical XOR immediate data with A	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2						

	mnemonic	function	description	instruction code								cycles	bytes	flags				
				D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1	BS
BRANCH	DJNZ Rr, addr	(Rr)←(Rr) −1 if (Rr) not zero: (PC_{0-7})←addr	Decrement the specified register and test contents	1 a_7	1 a_6	1 a_5	0 a_4	1 a_3	r a_2	r a_1	r a_0	2	2					
	JBb addr	(PC_{0-7})←addr if Bb = 1; (PC)←(PC) + 2 if Bb = 0	Jump to specified address if accumulator bit is set	b_2 a_7	b_1 a_6	b_0 a_5	1 a_4	0 a_3	0 a_2	1 a_1	0 a_0	2	2					
	JC addr	(PC_{0-7})←addr if C = 1; (PC)←(PC) + 2 if C = 0	Jump to specified address if carry flag is set	1 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JF0 addr	(PC_{0-7})←addr if (F0 = 1; (PC)←(PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set	1 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JF1 addr	(PC_{0-7})←addr if F1 = 1; (PC)←(PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set	0 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JMP addr	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11})←(DBF)	Direct jump to specified address within the 2K address block	a_{10} a_7	a_9 a_6	a_8 a_5	0 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2					
	JMPP @A	(PC_{0-7})←((A))	Jump indirect to specified address within address page	1	0	1	1	0	0	1	1	2	1					
	JNC addr	(PC_{0-7})←addr if C = 0; (PC)←(PC) + 2 if C = 1	Jump to specified address if carry flag is LOW	1 a_7	1 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNI	(PC_{0-7})←addr if $\overline{INT}$ = 0; (PC)←(PC) + 2 if $\overline{INT}$ = 1	Jump to specified address if $\overline{INT}$ input is LOW	1 a_7	0 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNT0 addr	(PC_{0-7})←addr if T0 = 0; (PC)←(PC) + 2 if T0 = 1	Jump to specified address if T0 is LOW	0 a_7	0 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNT1 addr	(PC_{0-7})←addr if T1 = 0; (PC)←(PC) + 2 if T1 = 1	Jump to specified address if T1 is LOW	0 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					

BRANCH (continued)	JNZ addr	(PC_{0-7})←addr if A ≠ 0; (PC)←(PC) + 2 if A = 0	Jump to specified address if A is non-zero	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JTF addr	(PC_{0-7})←addr if TF = 1; (PC)←(PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT0 addr	(PC_{0-7})←addr if T0 = 1; (PC)←(PC) + 2 if T0 = 0	Jump to specified address if T0 = 1	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT1 addr	(PC_{0-7})←addr if T1 = 1; (PC)←(PC) + 2 if T1 = 0	Jump to specified address if T1 = 1	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JZ addr	(PC_{0-7})←addr if A = 0; (PC)←(PC) + 2 if A ≠ 0	Jump to specified address if A is zero	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
CONTROL	EN I		Enable external ($\overline{INT}$) interrupt	0	0	0	0	0	1	0	1	1	1					
	DIS I		Disable external ($\overline{INT}$) interrupt	0	0	0	1	0	1	0	1	1	1					
	SEL RB0	(BS)←0	Select bank 0 (locations 0–7) of data memory	1	1	0	0	0	1	0	1	1	1					●
	SEL RB1	(BS)←1	Select bank 1 (locations 24–31) of data memory	1	1	0	1	0	1	0	1	1	1					●
	SEL MB0	(DBF)←0	Select program memory bank 0; addresses 0–2047	1	1	1	0	0	1	0	1	1	1					
	SEL MB1	(DBF)←1	Select program memory bank 1; addresses 2048–4095	1	1	1	1	0	1	0	1	1	1					
	ENT0 CLK		Enable clock output onto T0	0	1	1	1	0	1	0	1	1	1					

	mnemonic	function	description	instruction code								cycles	bytes	flags				
				D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1	BS
DATA MOVES	MOV A,#data	(A)←data	Move immediate data into A	0 d_7	0 d_6	1 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
	MOV A,Rr	(A)←(Rr) for r = 0–7	Move the contents of the designated register into A	1	1	1	1	1	r	r	r	1	1					
	MOV A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of data memory into A	1	1	1	1	0	0	0	r	1	1					
	MOV A,PSW	(A)←(PSW)	Move contents of the program status word into A	1	1	0	0	0	1	1	1	1	1					
	MOV Rr,#data	(Rr)←data for r = 0–7	Move immediate data into the designated register	1 d_7	0 d_6	1 d_5	1 d_4	1 d_3	r d_2	r d_1	r d_0	2	2					
	MOV Rr,A	(Rr)←(A) for r = 0–7	Move A contents into the designated register	1	0	1	0	1	r	r	r	1	1					
	MOV @Rr,A	((Rr))←(A) for r = 0–1	Move indirect A contents into data memory location	1	0	1	0	0	0	0	r	1	1					
	MOV @Rr,#data	((Rr))←data for r = 0–1	Move indirect the specified data into data memory	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	r d_0	2	2					
	MOV PSW,A	(PSW)←A	Move contents of A into the program status word	1	1	0	1	0	1	1	1	1	1	●	●	●		●
	MOVP A,@A	(A)←((A))	Move data in the current page into A	1	0	1	0	0	0	1	1	2	1					
	MOVP3 A,@A	(A)←((A)) in page 3	Move data in page 3 into A	1	1	1	0	0	0	1	1	2	1					
	MOVX A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of external memory location into A	1	0	0	0	0	0	0	r	2	1					
	MOVX @Rr,A	((Rr))←(A) for r = 0–1	Move indirect the contents of A into external memory	1	0	0	1	0	0	0	r	2	1					
	XCH A,Rr	(A)↔(Rr) for r = 0–7	Exchange A with designated register contents	0	0	1	0	1	r	r	r	1	1					
	XCH A,@Rr	(A)↔((R)) for r = 0–1	Exchange indirect A contents with location in data memory	0	0	1	0	0	0	0	r	1	1					

D.M.	XCHD A,@Rr	$(A_{0-3}) \leftrightarrow (Rr_{0-3})$ for r = 0–1	Exchange indirect 4-bit contents of A with data memory	0	0	1	1	0	0	0	r	1	1					
FLAGS	CPL C	(C)←NOT (C)	Complement content of carry bit	1	0	1	0	0	1	1	1	1	1	●				
	CPL F0	(F0)←NOT (F0)	Complement content of flag F0	1	0	0	1	0	1	0	1	1	1			●		
	CPL F1	(F1)←NOT (F1)	Complement content of flag F1	1	0	1	1	0	1	0	1	1	1				●	
	CLR C	(C)←0	Clear content of carry bit to 0	1	0	0	1	0	1	1	1	1	1	●				
	CLR F0	(F0)←0	Clear content of flag F0 to 0	1	0	0	0	0	1	0	1	1	1			●		
	CLR F1	(F1)←0	Clear content of flag F1 to 0	1	0	1	0	0	1	0	1	1	1				●	
INPUT/OUTPUT	ANL BUS,#data	(BUS)←(BUS) AND data	Logical AND immediate data with BUS	1	0	0	1	1	0	0	0	2	2					
				d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0							
	ANL Pp,#data	(Pp)←(Pp) AND data; p = 1–2	Logical AND immediate data with designated port (1 or 2)	1	0	0	1	1	0	p	p	2	2					
				d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0							
	ANLD Pp,A	(Pp)←(Pp) AND (A_{0-3}); p = 4–7	Logical AND contents of A with designated port (4–7)	1	0	0	1	1	1	p	p	2	1					
	IN A,Pp	(A)←(Pp) p = 1–2	Input data from designated port (1–2) into A	0	0	0	0	1	0	p	p	2	1					
	INS A,BUS	(A)←(BUS)	Input strobed BUS data into A	0	0	0	0	1	0	0	0	1	2					
	MOVD A,Pp	(A_{0-3})←(Pp); p = 4–7 (A_{4-7})←0	Move contents of designated port (4–7) into A	0	0	0	0	1	1	p	p	2	1					
	MOVD Pp,A	(Pp)←(A_{0-3}) p = 4–7	Move contents of A to designated port (4–7)	0	0	1	1	1	1	p	p	1	1					
	ORLD Pp,A	(Pp)←(Pp) OR (A_{0-3}); p = 4–7	Logical OR contents of A with designated port (4–7)	1	0	0	0	1	1	p	p	1	1					
	ORL BUS,#data	(BUS)←(BUS) OR data	Logical OR immediate data with BUS	1	0	0	0	1	0	0	0	2	2					
				d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0							
	ORL Pp,#data	(Pp)←(Pp) OR data p = 1–2	Logical OR immediate data with designated port (1–2)	1	0	0	0	1	0	p	p	2	2					
				d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0							
	OUTL BUS,A	(BUS)←(A)	Output contents A onto BUS	0	0	0	0	0	0	1	0	1	2					
	OUTL Pp,A	(Pp)←(A) p = 1–2	Output contents A to designated port (1–2)	0	0	1	1	1	0	p	p	1	1					

	mnemonic	function	description	instruction code								cycles	bytes	flags				
				D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1	BS
REGISTER	DEC Rr	(Rr)←(Rr) −1 for r = 0–7	Decrement contents of designated register by 1	1	1	0	0	1	r	r	r	1	1					
	INC Rr	(Rr)←(Rr) + 1 for r = 0–7	Increment contents of designated register by 1	0	0	0	1	1	r	r	r	1	1					
	INC @Rr	((R))←((Rr)) + 1 for r = 0–1	Increment indirect the contents of data memory location by 1	0	0	0	1	0	0	0	r	1	1					
SUBROUTINE	CALL addr	((SP))←(PC), (PSW_{4-7}) (SP)←(SP) + 1 $(PC_{8-10})\leftarrow addr_{8-10}$ $(PC_{0-7})\leftarrow addr_{0-7}$	Call designated subroutine	a_{10} a_7	a_9 a_6	a_8 a_5	1 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2					
	RET	(SP)←(SP) −1 (PC)←((SP))	Return from subroutine without restoring program status word	1	0	0	0	0	0	1	1	2	1					
	RETR	(SP)←(SP) −1 (PC)←((SP)) $(PSW_{4-7})\leftarrow((SP))$	Return from subroutine restoring program status word	1	0	0	1	0	0	1	1	2	1					
TIMER/COUNTER	EN TCNTI		Enable timer/counter interrupt	0	0	1	0	0	1	0	1	1	1					
	DIS TCNTI		Disable timer/counter interrupt	0	0	1	1	0	1	0	1	1	1					
	MOV A,T	(A)←(T)	Move contents of timer/counter into A	0	1	0	0	0	0	1	0	1	1					
	MOV T,A	(T)←(A)	Move contents of A into timer/counter	0	1	1	0	0	0	1	0	1	1					
	STOP TCNT		Stop count for event counter or timer	0	1	1	0	0	1	0	1	1	1					
	STRT CNT		Start count for event counter	0	1	0	0	0	1	0	1	1	1					
	STRT T		Start count for timer	0	1	0	1	0	1	0	1	1	1					
	NOP		No operation	0	0	0	0	0	0	0	0	1	1					

Table 3 Instruction timing (see also Figs 11 and 12)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	fetch instruction	increment program counter	–	increment timer	–	–	read port	• –	–	–
OUTL P,A	⋮	⋮	–	⋮	output to port	–	–	• –	–	–
ANL P,#data	⋮	• ⋮	–	⋮	read port	fetch immediate data	–	•increment program counter	output to port	–
ORL P,#data	⋮	• ⋮	–	⋮	read port	fetch immediate data	–	•increment program counter	output to port	–
INS A,BUS	⋮	⋮	–	⋮	–	–	read port	• –	–	–
OUTL BUS,A	⋮	⋮	–	⋮	output to port	–	–	• –	–	–
ANL BUS,#data	⋮	• ⋮	–	⋮	read port	fetch immediate data	–	•increment program counter	output to port	–
ORL BUS,#data	⋮	• ⋮	–	⋮	read port	fetch immediate data	–	•increment program counter	output to port	–
MOVX @R,A	⋮	⋮	output RAM address	⋮	output data to RAM	–	–	• –	–	–
MOVX A,@R	⋮	⋮	output RAM address	⋮	–	–	read data	• –	–	–
MOVD A,P	fetch instruction	increment program counter	output opcode/ address	increment timer	–	–	read P2 lower	• –	–	–

Continued on next page.

Table 2 Instruction timing (continued)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
MOVD P,A	fetch instruction	increment program counter	output opcode/ address	increment timer	output data to P2 lower	–	–	• –	–	–
ANLD P,A	¦		output opcode/ address	¦	output data	–	–	• –	–	–
ORLD P,A	¦		output opcode/ address	¦	output data	–	–	• –	–	–
J (conditional)	¦	•	sample condition	increment timer	–	fetch immediate data	–	•update program counter	–	–
STRT CNT STRT T	¦	•	–	–	start counter					
STOP TCNT	¦	•	–	–	stop counter					
EN I	¦	•	–	enable interrupt	–					
DIS I	¦	•	–	disable interrupt	–					
ENT0 CLK	fetch instruction	•increment program counter	–	enable clock	–					

- Valid instruction addresses are output at this time if external program memory is being accessed.

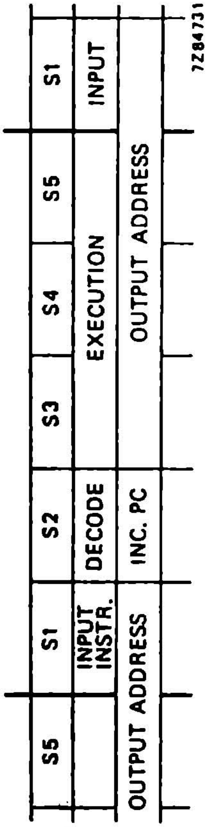

S5
S1
S2
S3
S4
S5
S1
INPUT INSTR.
DECODE
EXECUTION
INPUT
OUTPUT ADDRESS
INC. PC
OUTPUT ADDRESS
7Z84731

Fig. 11 Instruction cycle.

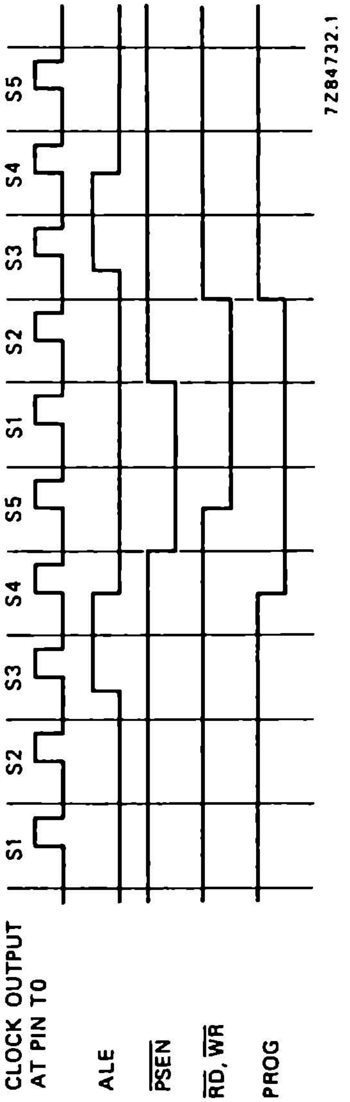

CLOCK OUTPUT AT PIN T0
ALE
$\overline{\text{PSEN}}$
$\overline{\text{RD}}$, $\overline{\text{WR}}$
PROG
S1
S2
S3
S4
S5
S1
S2
S3
S4
S5
7Z84732.1

Fig. 12 Instruction cycle timing.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage with respect to V_{SS}			
except input EA	V_I	–0,5 to + 7	V
input EA	V_I	–0,5 to + 15	V
D.C. current into any input or output	$\pm I_I$, $\pm I_O$	max. 10	mA
Total power dissipation	P_{tot}	max. 1,5	W
Storage temperature range	T_{stg}	–65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit	conditions
Supply voltage	$V_{CC} = V_{DD}$	4,5	5,0	5,5	V	
Supply current	I_{DD}	–	–	8	mA	
Supply current (total)	$I_{CC} + I_{DD}$	–	–	80	mA	
Inputs						
Input voltage LOW all inputs except XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IL}	–0,5	–	0,8	V	
Input voltage LOW XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IL1}	–0,5	–	0,6	V	
Input voltage HIGH all inputs except XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IH}	2,0	–	V_{CC}	V	
Input voltage HIGH XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IH1}	3,8	–	V_{CC}	V	
Outputs						
Output voltage LOW; BUS	V_{OL}	–	–	0,45	V	I_{OL} = 2 mA
Output voltage LOW $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE	V_{OL1}	–	–	0,45	V	I_{OL1} = 1,8 mA
Output voltage LOW; PROG	V_{OL2}	–	–	0,45	V	I_{OL2} = 1 mA
Output voltage LOW all other outputs	V_{OL3}	–	–	0,45	V	I_{OL3} = 1,6 mA
Output voltage HIGH; BUS	V_{OH}	2,4	–	–	V	$-I_{OH}$ = 400 µA
Output voltage HIGH $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE	V_{OH1}	2,4	–	–	V	$-I_{OH1}$ = 100 µA
Output voltage HIGH all other outputs	V_{OH2}	2,4	–	–		$-I_{OH2}$ = 40 µA

parameter	symbol	min.	typ.	max.	unit	conditions
Input leakage current T1, $\overline{\text{INT}}$	$\pm I_{IL}$	–	–	10	μA	$V_{SS} < V_I < V_{CC}$
Input leakage current P10 to P17, P20 to P27, EA, $\overline{\text{SS}}$	$-I_{IL1}$	–	–	500	μA	V_{SS} + 0,45 V $< V_I < V_{CC}$
Output leakage current BUS, T0 (high impedance)	$\pm I_{OL}$	–	–	10	μA	V_{SS} + 0,45 V $< V_I < V_{CC}$

A.C. CHARACTERISTICS

$V_{CC} = V_{DD} = 5\ V \pm 10\%$; $V_{SS} = 0\ V$; $T_{amb} = 0$ to $+ 70\ ^{o}C$

See waveforms Figs 13, 14 and 15

parameter	symbol	f = 6 MHz		f = 8 MHz		unit	conditions note 1
		min.	max.	min.	max.		
ALE pulse width	t_{LL}	400	–	270	–	ns	
Address set-up time to ALE	t_{AL}	75	–	75	–	ns	
Address hold time from ALE	t_{LA}	65	–	65	–	ns	
Control pulse width $\overline{\mathrm{PSEN}}$, $\overline{\mathrm{RG}}$, $\overline{\mathrm{WR}}$	t_{CC}	700	–	490	–	ns	
Data set-up time before $\overline{\mathrm{WR}}$	t_{DW}	370	–	370	–	ns	
Data hold time after $\overline{\mathrm{WR}}$	t_{WD}	80	–	80	–	ns	note 2
Cycle time	t_{CY}	2,5	–	1,875	–	μs	
Data hold time	t_{DR}	0	200	0	150	ns	
$\overline{\mathrm{PSEN}}$, $\overline{\mathrm{RD}}$ to data input	t_{RD}	–	500	–	340	ns	
Address set-up time to $\overline{\mathrm{WR}}$	t_{AW}	230	–	210	–	ns	
Address set-up time to data input	t_{AD}	–	950	–	650	ns	
Address floating to $\overline{\mathrm{RD}}$, $\overline{\mathrm{PSEN}}$	t_{AFC}	0	–	0	–	ns	
Control pulse to ALE	t_{CA}	10	–	10	–	ns	

Notes

1. Control outputs: $C_L = 80$ pF
   Bus outputs: $C_L = 150$ pF
2. Bus high-impedance load: 20 pF

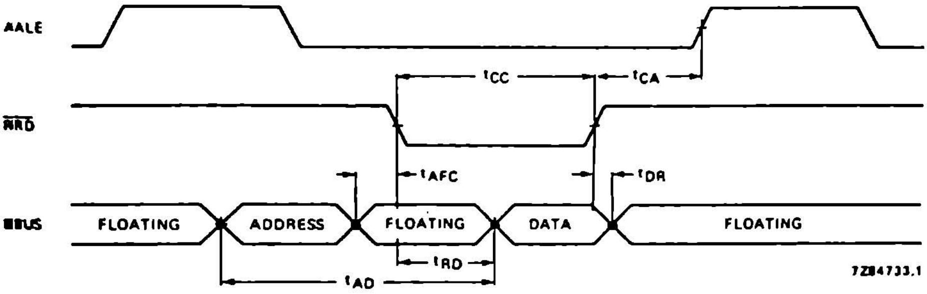

ALE
RD
BUS
tCC
tCA
tAFC
tDR
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
tRD
tAD
7Z84733.1

Fig. 13 Read from external data memory.

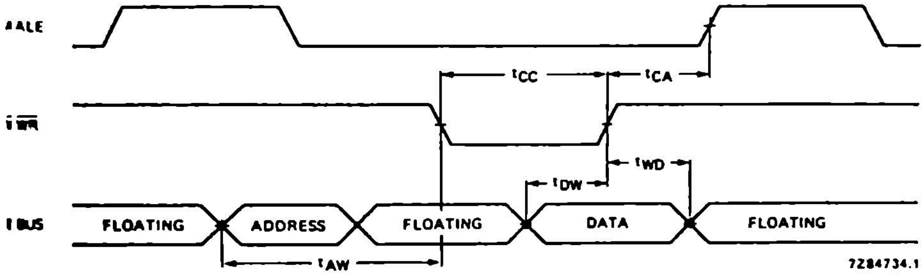

ALE
WR
BUS
tCC
tCA
tWD
tDW
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
tAW
7Z84734.1

Fig. 14 Write to external data memory.

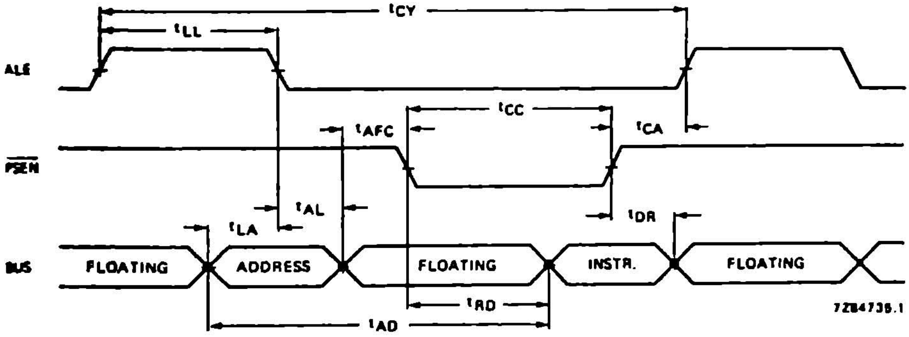

tCY
tLL
ALE
tCC
tAFC
tCA
PSEN
tAL
tLA
tDR
BUS
FLOATING
ADDRESS
FLOATING
INSTR.
FLOATING
tRD
tAD
7Z84735.1

Fig. 15 Instruction fetch from external program memory.

A.C. CHARACTERISTICS

Port 2 timing (see Fig. 16)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C

parameter	symbol	f = 6 MHz min.	f = 6 MHz max.	f = 8 MHz min.	f = 8 MHz max.	unit
Port control set-up time before falling edge of PROG	t_{CP}	110	–	105	–	ns
Port control hold time after falling edge of PROG	t_{PC}	100	–	90	–	ns
PROG to time P_2 input must be valid	t_{PR}	–	810	–	700	ns
Input data hold time	t_{PF}	0	150	0	150	ns
Output data set-up time	t_{DP}	250	–	210	–	ns
Output data hold time	t_{PD}	65	–	35	–	ns
PROG pulse width	t_{PP}	1200	–	970	–	ns
Port 2 I/O data set-up time	t_{PL}	350	–	300	–	ns
Port 2 I/O data hold time	t_{LP}	150	–	65	–	ns
Control pulse to ALE	t_{CA}	10	–	10	–	ns

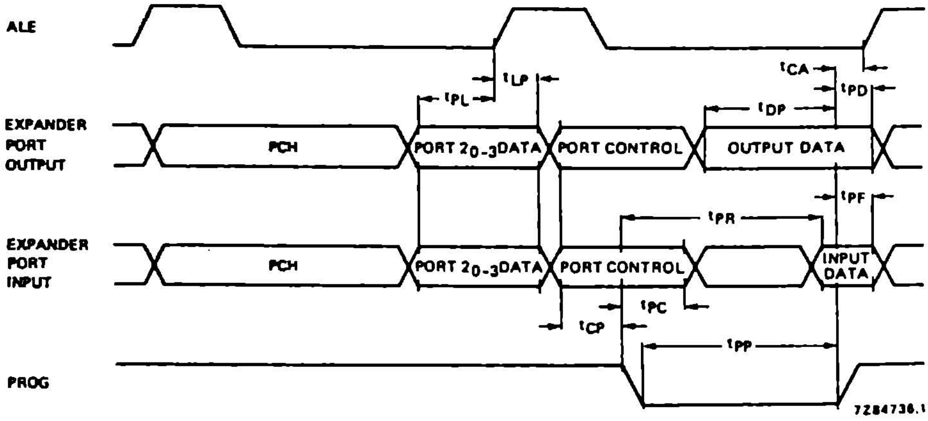
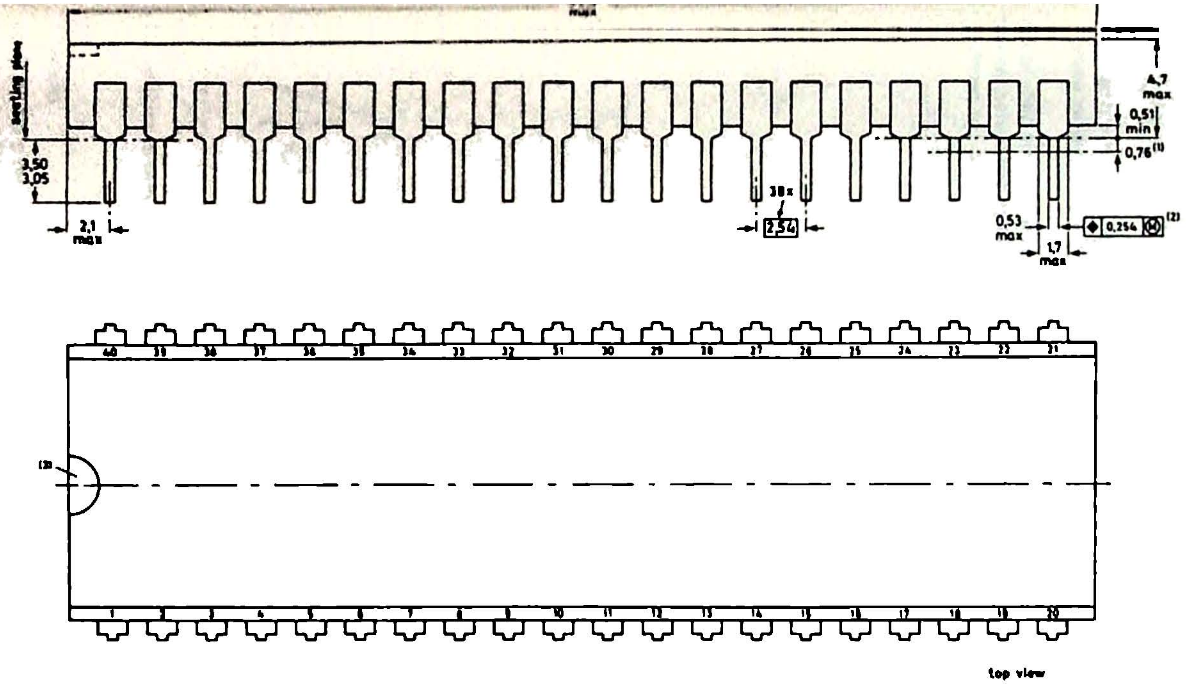
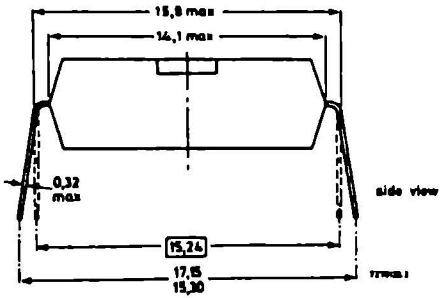

ALE
tLP
tPL
tCA
tPD
tDP
EXPANDER PORT OUTPUT
PCH
PORT 20-3DATA
PORT CONTROL
OUTPUT DATA
tPF
tPR
EXPANDER PORT INPUT
PCH
PORT 20-3DATA
PORT CONTROL
INPUT DATA
tPC
tCP
tPP
PROG
7Z84736.1

Fig. 16 Port 2 timing.

40-LEAD DUAL IN-LINE; PLASTIC (SOT-129)


4,7 max
0,51 min
0,76 (1)
0,254 (2)
0,53 max
1,7 max
38x
2,54
2,1 max
3,50
3,05
seating plane
max
(3)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
top view


15,8 max
14,1 max
0,32 max
15,24
17,15
15,90
side view

Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

SOLDERING

See next page.

SOLDERING

1. By hand

Apply the soldering iron below the seating plane (or not more than 2 mm above it).
If its temperature is below 300 °C it must not be in contact for more than 10 seconds; if between 300 °C and 400 °C, for not more than 5 seconds.

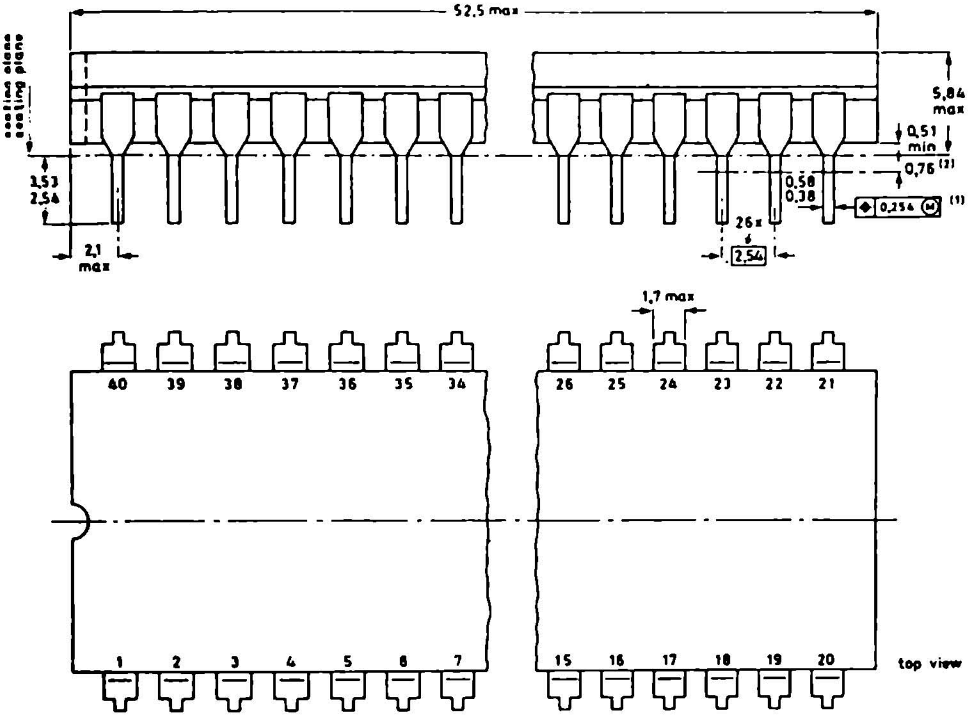
2. By dip or wave

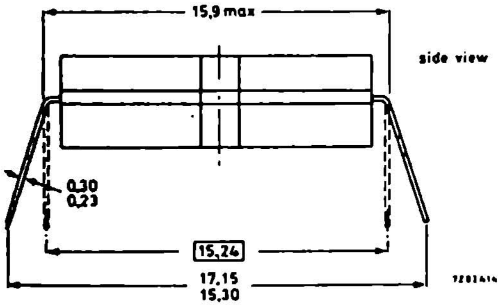
The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.
The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

3. Repairing soldered joints

The same precautions and limits apply as in (1) above.

40-LEAD DUAL IN-LINE; CERAMIC (SOT-145)


52,5 max
seating plane
5,84 max
0,51 min
0,76 (2)
0,58
0,38
0,254 M (1)
3,53
2,54
2,1 max
26x
2,54
1,7 max
40 39 38 37 36 35 34
26 25 24 23 22 21
1 2 3 4 5 6 7
15 16 17 18 19 20
top view


15,9 max
side view
0,30
0,23
15,24
17,15
15,30

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

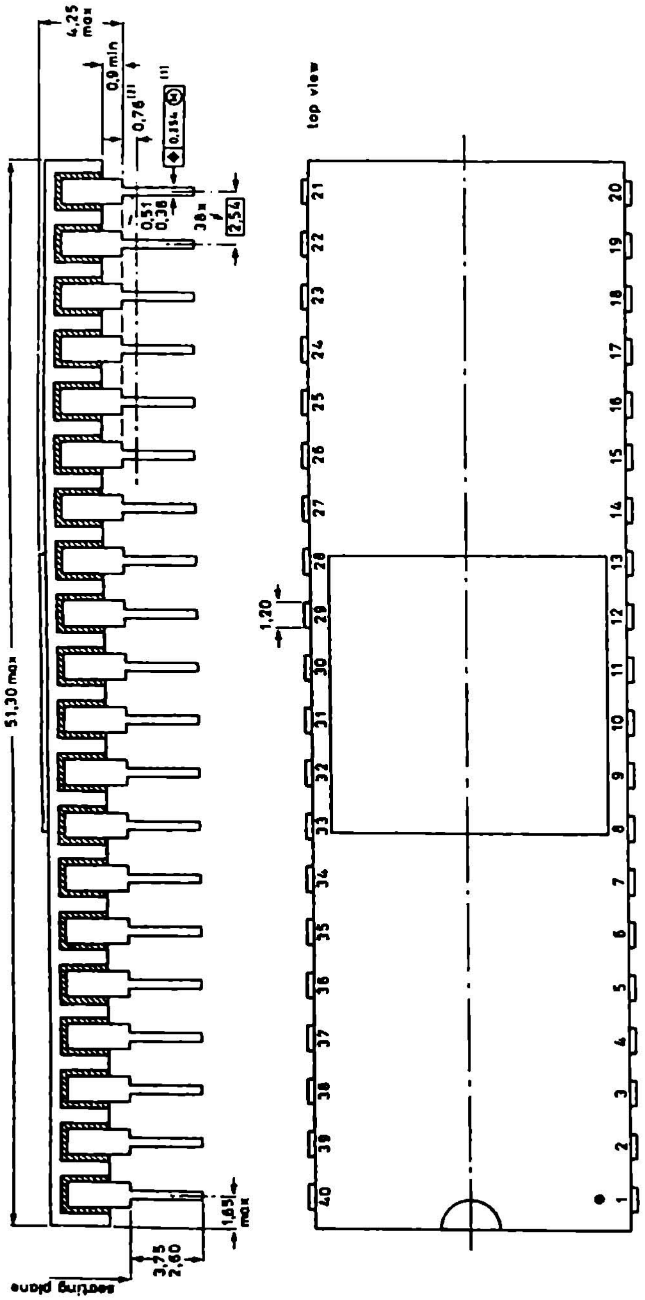
(2) Lead spacing tolerances apply from seating plane to the line indicated.

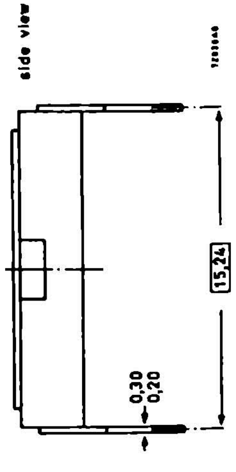
Dimensions in mm

Remarks

1. Leads are given positive misalignment so that they grip after insertion.
2. Leads are Ni-Fe, pure tin plated.

40-LEAD DUAL IN-LINE; METAL-CERAMIC (SOT-88B)


4,25 max
0,9 min
0,76 (1)
0,254 Ⓜ (1)
0,51
0,38
38×
2,54
51,30 max
1,65 max
3,75
2,60
seating plane
top view
1,20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1


side view
0,30
0,20
15,24

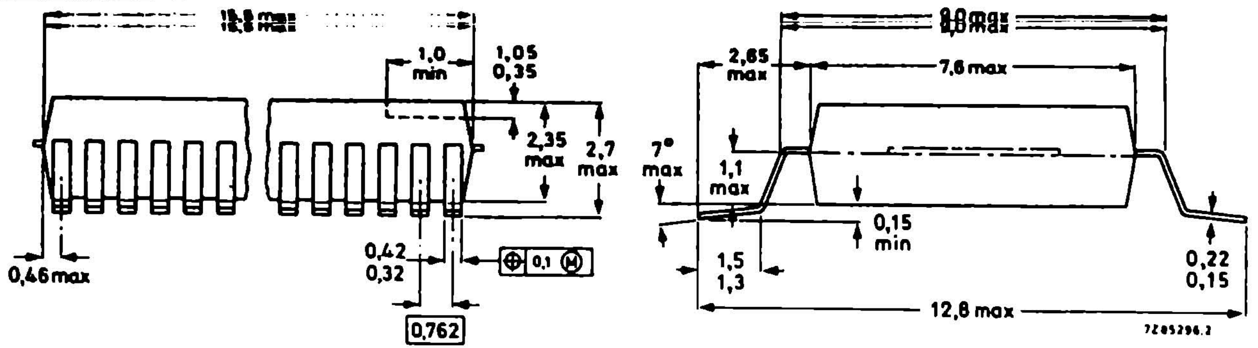
⊕ Positional accuracy.

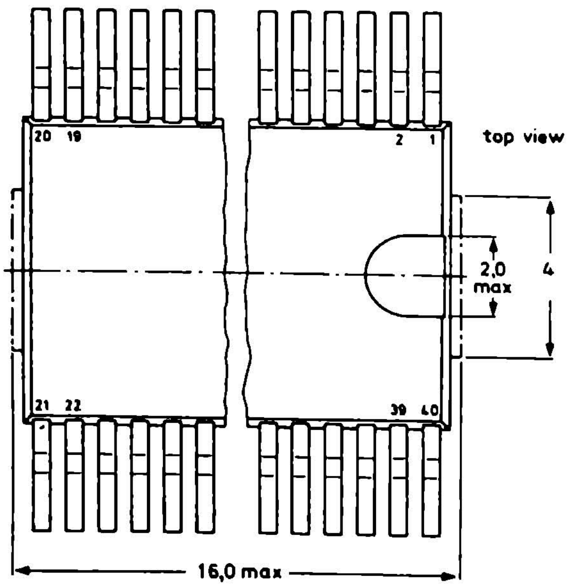
Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

40-LEAD FLAT PACK; PLASTIC (VSO-40; SOT-158)


1,0
min
1,05
0,35
2,35
max
2,7
max
0,46 max
0,42
0,32
0,1 Ⓜ
0,762
2,65
max
7,6 max
7°
max
1,1
max
0,15
min
1,5
1,3
0,22
0,15
12,8 max
7Z85296.2


top view
20
19
2
1
21
22
39
40
2,0
max
4
16,0 max

Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

SOLDERING

See next page.

TEST SOCKET

Catalogue no.: 7332 150 07601

SOLDERING

1. Soldering iron or pulse heated solder tool

Apply the heating tool to the flat part of the pin only.
Limit the contact time to maximum 10 seconds up to 300 °C, or 5 seconds up to maximum 400 °C.
When using the proper tools, up to 20 pins (at one side of the device) can be soldered in one operation with 2 to 5 seconds and 270 to 320 °C.

2. By dip or wave

The maximum permissible temperature of the solder is 260 °C. The permissible total time of immersing the whole package in the bath is 10 seconds, if it is allowed to cool down to less than 150 °C within 6 seconds.

3. Repairing soldered joints

The same precautions and limits apply as in (1) above.
If the vertical part of the pin needs heating, reduce the soldering iron temperature to 260 °C.

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The MAB8049H family of single-chip 8-bit microcomputers are fabricated in H-MOS.
Two interchangeable (pin compatible) versions are available:

- The MAB8049H with resident mask-programmed ROM,
- The MAB8039H without resident program memory for use with external EPROM/ROM.

The MAB8049H family are designed to be efficient control processors as well as arithmetic processors. Their instruction set allows the user to directly set and reset individual I/O lines as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions enable efficient implementation of standard logic functions. Code efficiency is high; over 70% of the instructions are single byte; all others are two byte.

An on-chip 8-bit counter is provided, which can count either machine cycles (÷ 32) or external events. The counter can be programmed to cause an interrupt to the processor.
Program and data memories can be expanded using standard devices. Input/output capabilities can be expanded using standard devices.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 40-pin package
- 2K x 8 ROM, 128x8 RAM, 27 I/O lines
- Internal counter/timer
- Internal oscillator, clock driver
- Single-level interrupts: external and counter/timer
- 17 internal registers: accumulator, 16 addressable registers
- Over 90 instructions: 70% single byte
- All instructions: 1 or 2 cycles (1,36/2,5 μs per cycle)
- Easily expandable memory and I/O
- TTL compatible inputs and outputs
- Single 5 V supply

APPLICATIONS

- Peripheral interfaces and controllers
- Test and measurement instruments
- Sequencers
- Audio/video systems
- Environmental control systems
- Modems and data enciphering

PACKAGE OUTLINES

MAB8049HP: 40-lead DIL; plastic (SOT-129).
MAB8049HD: 40-lead DIL; ceramic (SOT-145).
MAB8049HD: 40-lead DIL; metal-ceramic (SOT-88B).
MAB8049HT: 40-LEAD FLAT PACK; PLASTIC (VSO-40; SOT-158)

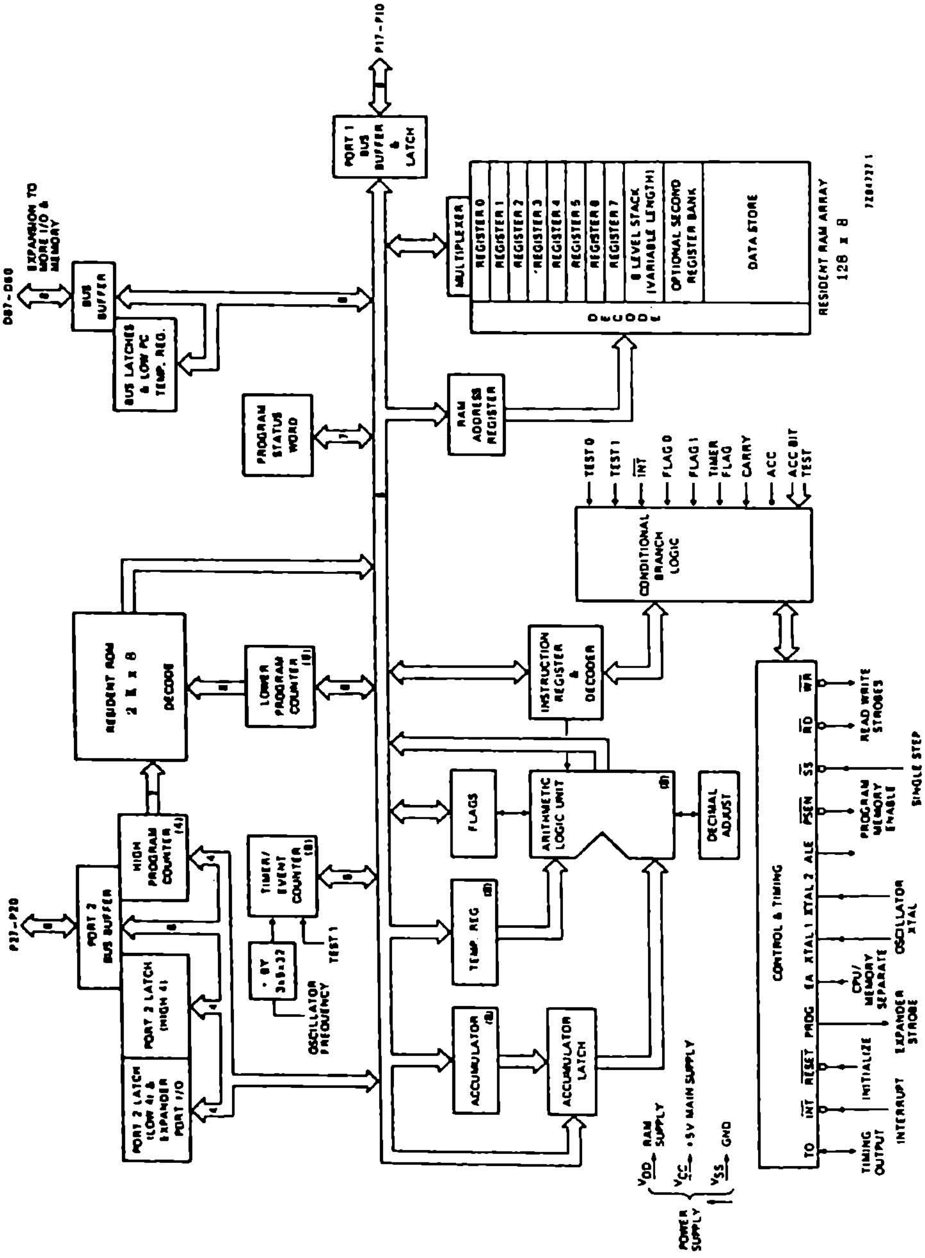
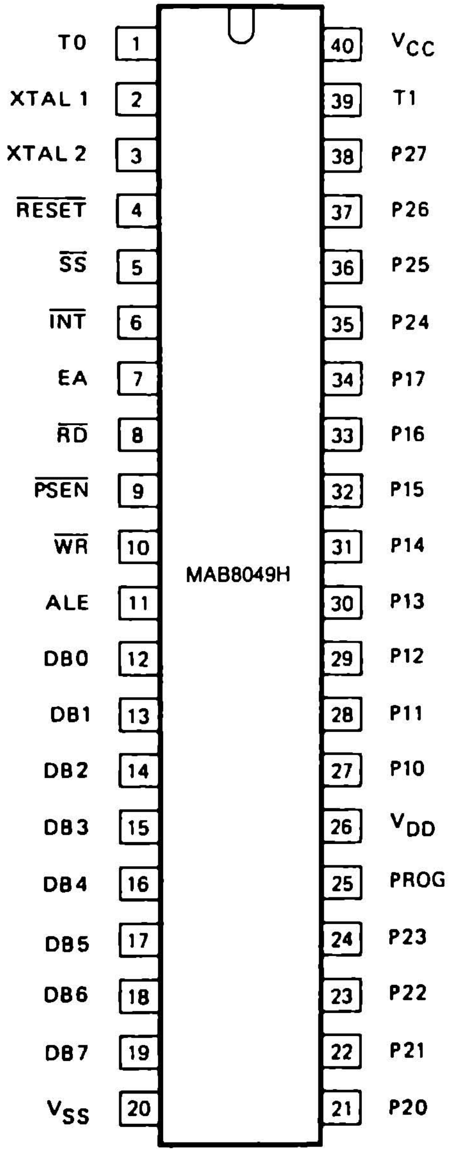

P17–P10
PORT 1 BUS BUFFER & LATCH
DB7–DB0
EXPANSION TO MORE I/O & MEMORY
BUS BUFFER
BUS LATCHES & LOW PC TEMP. REG.
PROGRAM STATUS WORD
MULTIPLEXER
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
8 LEVEL STACK (VARIABLE LENGTH)
OPTIONAL SECOND REGISTER BANK
DATA STORE
DECODE
RESIDENT RAM ARRAY 128 x 8
RAM ADDRESS REGISTER
TEST 0
TEST 1
INT
FLAG 0
FLAG 1
TIMER FLAG
CARRY
ACC
ACC BIT TEST
CONDITIONAL BRANCH LOGIC
RESIDENT ROM 2 K x 8
DECODE
LOWER PROGRAM COUNTER (8)
INSTRUCTION REGISTER & DECODER
P27–P20
PORT 2 BUS BUFFER
HIGH PROGRAM COUNTER (4)
PORT 2 LATCH (HIGH 4)
PORT 2 LATCH (LOW 4) & EXPANDER PORT I/O
TIMER/ EVENT COUNTER (8)
÷ BY 3x5x32
OSCILLATOR FREQUENCY
TEST 1
FLAGS
ARITHMETIC LOGIC UNIT (8)
DECIMAL ADJUST
TEMP. REG (8)
ACCUMULATOR (8)
ACCUMULATOR LATCH
VDD RAM SUPPLY
VCC +5V MAIN SUPPLY
VSS GND
POWER SUPPLY
CONTROL & TIMING
TO
INT
RESET
PROG
EA
XTAL 1
XTAL 2
ALE
PSEN
SS
RD
WR
TIMING OUTPUT
INTERRUPT
INITIALIZE
EXPANDER STROBE
CPU/ MEMORY SEPARATE
OSCILLATOR XTAL
PROGRAM MEMORY ENABLE
SINGLE STEP
READ WRITE STROBES
8
7284727.1

Fig. 1 Block diagram.


T0
1
XTAL 1
2
XTAL 2
3
$\overline{RESET}$
4
$\overline{SS}$
5
$\overline{INT}$
6
EA
7
$\overline{RD}$
8
$\overline{PSEN}$
9
$\overline{WR}$
10
ALE
11
DB0
12
DB1
13
DB2
14
DB3
15
DB4
16
DB5
17
DB6
18
DB7
19
V_{SS}
20
MAB8049H
40
V_{CC}
39
T1
38
P27
37
P26
36
P25
35
P24
34
P17
33
P16
32
P15
31
P14
30
P13
29
P12
28
P11
27
P10
26
V_{DD}
25
PROG
24
P23
23
P22
22
P21
21
P20

7Z84742.1

Fig. 2 Pinning diagram; for pin designation see next page.

PIN DESIGNATION

designation	pin no.	function
DB0—DB7	12—19	BUS. Bidirectional I/O port that can be read or written using the $\overline{RD}$ and $\overline{WR}$ strobes. This port can also be statically latched. Contains the 8 lower order address bits during external memory access and receives the addressed instruction under control of $\overline{PSEN}$. $\overline{PSEN}$, ALE, $\overline{RD}$ and $\overline{WR}$ determine whether the access is an instruction fetch or a read/write access to external RAM.
P10—P17	27—34	**Port 1.** 8-bit quasi-bidirectional I/O port (note 1).
P20—P27	21—24, 35—38	**Port 2.** 8-bit quasi-bidirectional I/O port (note 1). P20—P23 contain the 4 higher order address bits during an access of external program memory and also serve as a 4-bit I/O expander bus for the 8243.
PROG	25	**Output strobe** (active LOW) for 8243 I/O expander.
T0	1	**Input pin** sensed using the JT0 and JNT0 instructions. **Clock output pin** when designated as such by the ENT0 CLK instruction.
T1	39	**Input pin** sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
$\overline{INT}$	6	**Interrupt input pin.** When LOW causes an interrupt in the current program if external interrupt is enabled. Can also be used as an input, testable using the JNI instruction. Interrupt is disabled during and after a RESET.
$\overline{RESET}$	4	**Reset input** pin used to initialize the microcomputer. Active LOW. During program verification the address is latched by a '0' to '1' transition on $\overline{RESET}$ and the data at the addressed location is output on BUS (note 2).
ALE	11	**Address latch enable.** Occurs each clock cycle and is useful for timing and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0 to DB7 outputs.
$\overline{RD}$	8	**Read** BUS. Active LOW strobe used to gate data onto BUS lines when reading from an external source.
$\overline{WR}$	10	**Write** BUS. Active LOW strobe used to write data from BUS lines to an external designation.
EA	7	**External access input.** When HIGH, forces instruction fetch from external memory.
$\overline{PSEN}$	9	**Program store enable.** Active LOW strobe that occurs only during a fetch from external program memory.
$\overline{SS}$	5	**Single step input.** Active LOW which is used with ALE to cause the microcomputer to execute a single instruction.
XTAL1	2	**One side of crystal** (or inductor) input for internal oscillator. Can also be used as an input for an external timing source (note 2).

designation	pin no.	function
XTAL2	3	Other side of crystal.
V_{SS}	20	Ground.
V_{CC}	40	Power supply, + 5 V.
V_{DD}	26	RAM standby power supply, low power pin.

Notes

1. Each port line can be designated as an input or an output. A line is designated as an input by first writing a logic '1' to the line. $\overline{\text{RESET}}$ sets all lines to logic '1'.
2. Non-standard TTL V_{IH}.

FUNCTIONAL DESCRIPTION

Detailed information available on request.

Program memory

The resident program memory consists of a 2048 byte ROM (MAB8049H); the MAB8039H has no resident program memory.
The total addressing capability is 4096 bytes.
The program memory address space is divided into two 2048-bytes banks MB0 and MB1 (Fig. 3).
Further, the program memory is divided into pages of 256 bytes each. This latter division applies only for conditional branches.
There are three locations in program memory of special importance. These locations contain the first instruction to be executed after one of three events.
The three locations and their contents are:

location 0 — activation, then deactivation of the $\overline{\text{RESET}}$ line.
location 3 — activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled,
location 7 — an overflow of the timer/counter if the T/C interrupt is enabled.

Data memory

The resident data memory, as shown in Fig. 4, consists of a128byte RAM. All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1. The first eight locations of RAM (0 to 7) are designated as working registers and are directly addressable by several instructions.
By selecting register bank 1, RAM locations 24 to 31 become the working registers, replacing those in register bank 0 (0 to 7).
RAM locations 8 to 23 are designated as the stack. Two locations (bytes) are used per CALL, allowing up to eight levels of subroutine nesting.
If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM (greater than 384 bytes total) is required, an I/O port can be used to select one (256 byte) bank of external memory at a time.

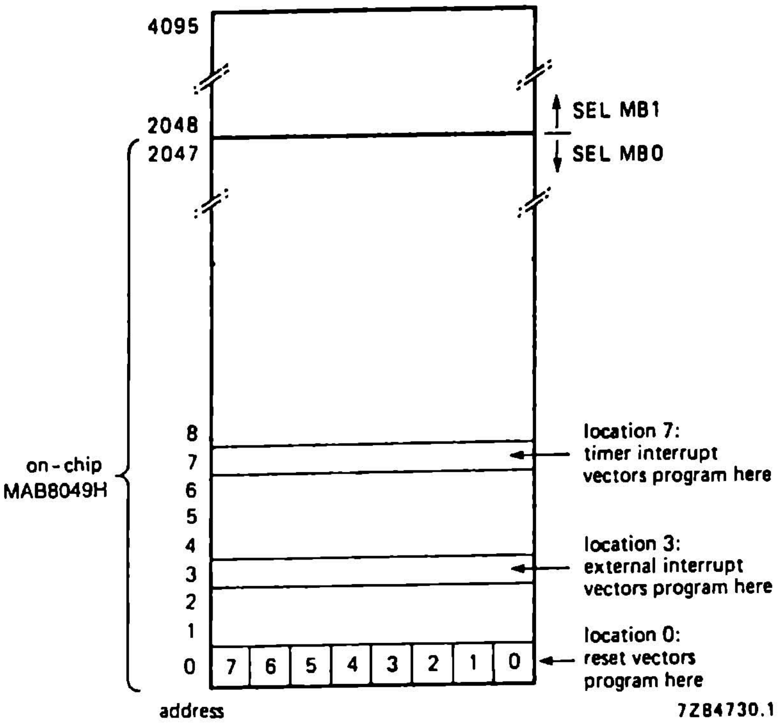

4095
2048
2047
SEL MB1
SEL MB0
8
7
6
5
4
3
2
1
0
7 6 5 4 3 2 1 0
address
on-chip MAB8049H
location 7: timer interrupt vectors program here
location 3: external interrupt vectors program here
location 0: reset vectors program here
7Z84730.1

Fig. 3 Program memory map.

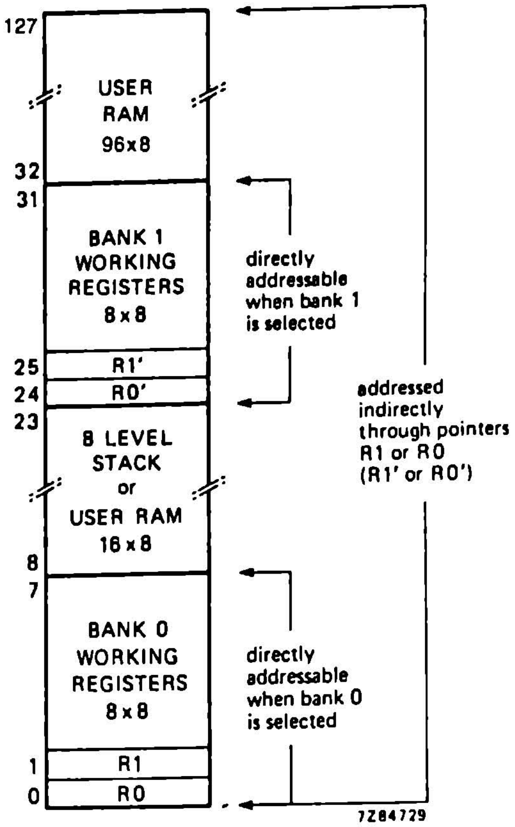

127
USER RAM 96x8
32
31
BANK 1 WORKING REGISTERS 8x8
25
R1'
24
R0'
23
8 LEVEL STACK or USER RAM 16x8
8
7
BANK 0 WORKING REGISTERS 8x8
1
R1
0
R0
directly addressable when bank 1 is selected
directly addressable when bank 0 is selected
addressed indirectly through pointers R1 or R0 (R1' or R0')
7Z84729

Fig. 4 Data memory map. In addition R0 or R1 (R0' or R1') may be used to address 256 words of an external RAM.

Program counter and stack

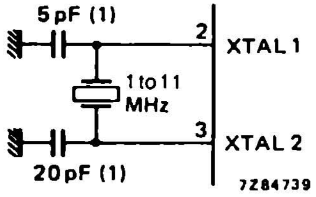
The program counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. When EA is '0', the PC can address locations 0 to 1023 of internal program memory. At the 1 K boundary, an automatic switch-over to external memory is made. When EA is '1', all the program is fetched from external ROM/EPROM. The total address space is 4 K bytes.

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the program status word (PSW). Data RAM locations 8 to 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000B, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Oscillator and clock

The MAB8049H contains its own internal oscillator and clock driver. A crystal, inductor or external pulse generator determines the oscillator frequency (see Figs 5, 6 and 7). The output of the oscillator is divided-by-three and is available at T0 (pin 1) by executing the ENT0 CLK instruction. This CLK signal is divided-by-five to define a machine (instruction) cycle. It is available at ALE (pin 11).



(1) Including crystal-socket stray capacitance.

Fig. 5 Crystal oscillator mode. Crystal series impedance should be < 75 Ω at 6 MHz and < 180 Ω at 3,6 MHz.

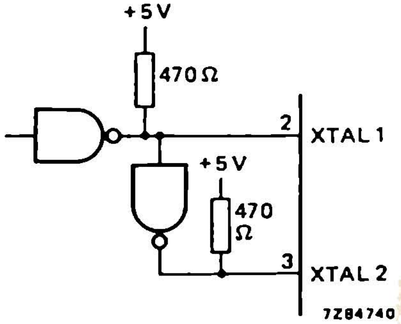
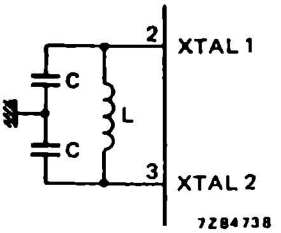


Fig. 6 Driving from external source. Both XTAL1 and XTAL2 should be driven. Resistors to V_{CC} (+ 5 V) are needed to ensure V_{IH} = 3,8 V if TTL circuitry is used. The minimum HIGH and LOW times are 45%.



L (µH)	C (pF)	nom. f (MHz)
45	20	5,2
120	20	3,2

$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3C_{pp}}{2}$$

Fig. 7 LC oscillator mode. Each C should be ≈ 20 pF including stray capacitance.
C_{pp} ≈ 5 to 10 pF (pin-to-pin capacitance).

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided-by-32 before they are applied to the input of the 8-bit counter. External events are applied directly to the input of the counter. The maximum frequency that can be counted is one third of the machine cycle frequency. The minimum positive duty cycle that can be detected is 0,2 times the cycle period. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

Interrupt

An interrupt may be generated by either an external input ($\overline{\text{INT}}$, pin 6) or the overflow of the internal counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine. After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

Input/output

The MAB8049H has 27 I/O lines. These lines are arranged as three 8-line ports, which serve as either inputs, outputs or bidirectional ports and 3 'test' inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, e.g., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.
The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The circuit configuration is shown in Fig. 8. Each line is continuously pulled up to + 5 V through a relative high resistance (≈ 50 kΩ). This pull-up is sufficient to provide the source current for a TTL HIGH level, yet can be pulled LOW by a standard TTL gate, thus allowing the same pin to be used for both input and output. To provide fast switching times during '0' to '1' transition a relatively low impedance transistor (≈ 5 kΩ) is switched on momentarily for 1/5 of a machine cycle whenever a '1' is written to the line. When a '0' is written, a low impedance (≈ 300 Ω) transistor overcomes the light pull-up and provides TTL current sinking capability.
Since the pull-down transistor is a low-impedance device, a '1' must first be written to any line which is to be used as an input. RESET initializes all lines to the high impedance '1' state. This structure allows input and output on the same pin and also allows a mixture of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

BUS

BUS is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.
As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{\text{RD}}$ and $\overline{\text{WR}}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{\text{WR}}$ output line and output data is valid at the trailing-edge of $\overline{\text{WR}}$. A read of the port generates a pulse on the $\overline{\text{RD}}$ output line and input data must be valid at the trailing-edge of $\overline{\text{RD}}$. When not being written or read, the BUS lines are high impedance.

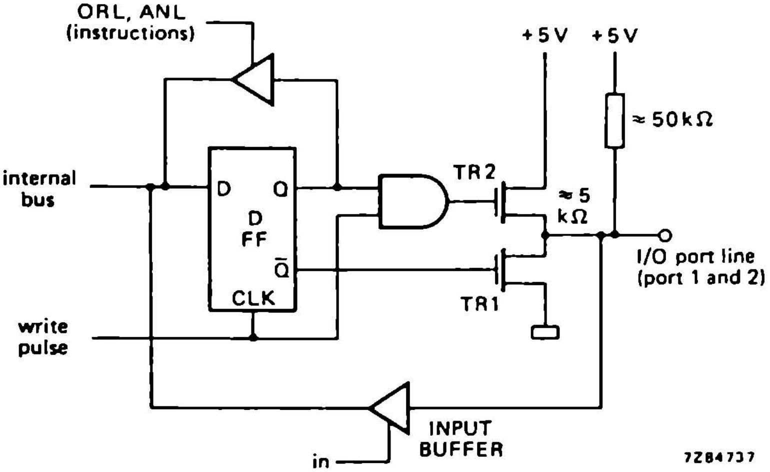


Fig. 8 Quasi-bidirectional port structure.

Test (T0, T1) and $\overline{INT}$ inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These pins are T0, T1, and $\overline{\text{INT}}$ and they allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1 and $\overline{\text{INT}}$ pins have other possible functions as well.

$\overline{\text{RESET}}$ input

The $\overline{\text{RESET}}$ input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pull-up resistor which, in combination with an external 1 μF capacitor, provides an internal reset pulse of sufficient duration to guarantee all circuitry is reset. If the reset pulse is generated externally, the $\overline{\text{RESET}}$ pin must be held at ground (0,45 V) for at least 10 ms after the power supply is within tolerance. Only five machine cycles (12,5 μs at 6 MHz) are required if power is already on and the oscillator has stabilized. Typical circuitry is shown in Fig. 9.

Single step input ($\overline{\text{SS}}$)

By proper control of the $\overline{\text{SS}}$ line, the processor can be forced to execute one instruction and then to wait until the single step switch is activated again.

Power-down mode

In the MAB8049H and MAB8039H , power can be removed from all but the 128x8 data RAM array, for low power standby operation. In the power-down mode the contents of the data RAM can be maintained while drawing typically 10% to 15% of the normal operating power supply voltage.
V_{CC} serves as the + 5 V supply pin for the bulk of the circuitry, while the V_{DD} pin supplies only the RAM array. In normal operation, both pins are at + 5 V. In standby, V_{CC} is at ground and only V_{DD} is maintained at + 5 V.
Applying $\overline{\text{RESET}}$ to the processor through the $\overline{\text{RESET}}$ pin inhibits any access to the RAM by the processor and guarantees that the RAM cannot be inadvertently altered as power is removed from V_{CC}. A typical power down sequence occurs as shown in Fig. 10.

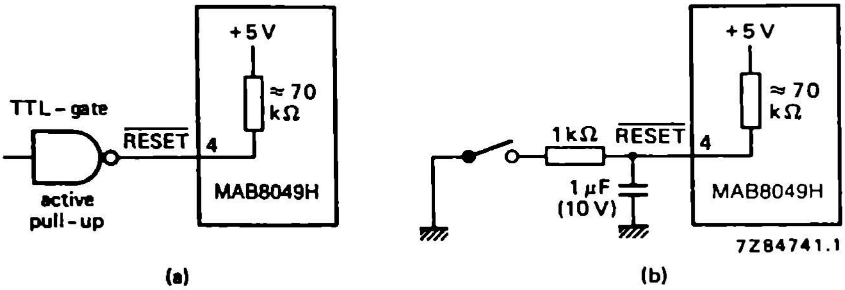

+5V
≈70 kΩ
TTL-gate
RESET
4
MAB8049H
active pull-up
(a)
+5V
≈70 kΩ
1kΩ
RESET
4
1 μF (10 V)
MAB8049H
7Z84741.1
(b)

Fig. 9 An external reset circuit is shown in (a) and power-on reset in (b).

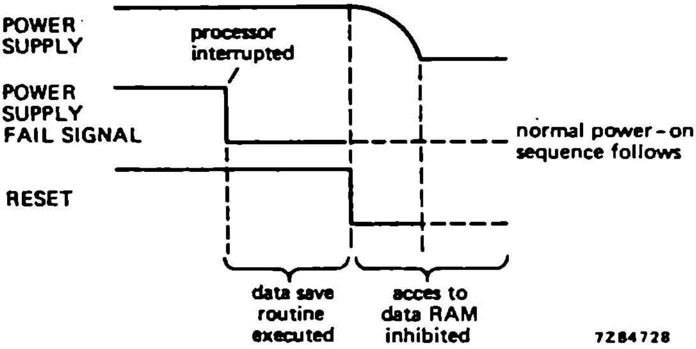

POWER SUPPLY
processor interrupted
POWER SUPPLY FAIL SIGNAL
normal power-on sequence follows
RESET
data save routine executed
acces to data RAM inhibited
7Z84728

Fig. 10 Power down sequence.

Instruction set

The MAB8049H instruction set consists of over 90 one and two-byte instructions (see Table 1). Program code efficiency is high because:

- working registers and program variables are stored in the RAM locations 0 to 127, which require only a single byte to address,
- program memory is divided into pages of 256 bytes, which means that branch destination addresses require one byte.

The instruction set efficiently manipulates and tests bits in addition to performing logical and arithmetic operations upon and the testing of bytes. A set of MOVE instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi-way branch (up to 256) upon the content of the accumulator to addresses stored in a look-up table. The 'decrement register and jump if not zero' instruction saves a byte every time it is used as opposed to using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The MAB8049H can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are highly desirable for real-time applications. See Table 2 for instruction timing.

Table 1 Instruction set is shown on the next 7 pages.

Symbols definitions used in Table 1.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number or expression (8-bits)
F0, F1	flags 0 and 1
I	interrupt
$\overline{\text{INT}}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1,	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
$	current value of program counter
←	is replaced by
↔	is exchanged with

Notes to Table 1.

1. Instruction code designations r and p form the binary representation of the registers and ports involved.
2. The dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

	mnemonic	function	description	instruction code								cycles	bytes	flags				
				D7	D6	D5	D4	D3	D2	D1	D0			C	AC	F0	F1	BS
ACCUMULATOR	ADD A,Rr	(A)←(A) + (Rr) for r = 0–7	Add contents of designated register to A	0	1	1	0	1	r	r	r	1	1	●	●			
	ADD A,@Rr	(A)←(A) + ((Rr)) for r = 0–1	Add indirect the contents of the data memory location to A	0	1	1	0	0	0	0	r	1	1	●	●			
	ADD A,#data	(A)←(A) + data	Add immediate data to A	0 d7	0 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	2	2	●	●			
	ADDC A,Rr	(A)←(A) + (C) + (Rr) for r = 0–7	Add with carry the contents of designated register to A	0	1	1	1	1	r	r	r	1	1	●	●			
	ADDC A,@Rr	(A)←(A) + (C) + ((Rr)) for r = 0–1	Add indirect with carry the contents of the data memory location to A	0	1	1	1	0	0	0	r	1	1	●	●			
	ADDC A,#data	(A)←(A) + (C) + data	Add immediate data with carry to A	0 d7	0 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	2	2	●	●			
	ANL A,Rr	(A)←(A) AND (Rr) for r = 0–7	Logical AND contents of designated register with A	0	1	0	1	1	r	r	r	1	1					
	ANL A,@Rr	(A)←(A) AND ((Rr)) for r = 0–1	Logical AND indirect the contents of data memory with A	0	1	0	1	0	0	0	r	1	1					
	ANL A,#data	(A)←(A) AND data	Logical AND immediate data with A	0 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	2	2					
	CLR A	(A)←0	Clear the contents of A	0	0	1	0	0	1	1	1	1	1					
	CPL A	(A)←NOT(A)	Complement the contents of A	0	0	1	1	0	1	1	1	1	1					
	DA A		Decimal adjust the contents of A	0	1	0	1	0	1	1	1	1	1	●	●			
	DEC A	(A)←(A) –1	Decrement the contents of A by 1	0	0	0	0	0	1	1	1	1	1					
	INC A	(A)←(A) +1	Increment the contents of A by 1	0	0	0	1	0	1	1	1	1	1					
	ORL A,Rr	(A)←(A) OR (Rr) for r = 0–7	Logical OR contents of designated register with A	0	1	0	0	1	r	r	r	1	1					
	ORL A,@Rr	(A)←(A) OR ((Rr)) for r = 0–1	Logical OR indirect the contents of data memory location with A	0	1	0	0	0	0	0	r	1	1					

ACCUMULATOR (continued)	ORL A,#data	(A)←(A) OR data	Logical OR Immediate data with A	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2				
	RL A	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (A_7)$ for n = 0–6	Rotate A left by 1-bit without carry	1	1	1	0	0	1	1	1	1	1				
	RLC A	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$ for n = 0–6	Rotate A left by 1-bit through carry	1	1	1	1	0	1	1	1	1	1	●			
	RR A	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$ for n = 0–6	Rotate A right by 1-bit without carry	0	1	1	1	0	1	1	1	1	1				
	RRC A	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0)$ n = 0–6	Rotate A right by 1-bit through carry	0	1	1	0	0	1	1	1	1	1	●			
	SWAP A	$(A_{4-7}) \leftrightarrow (A_{0-3})$	Swap the two 4-bit nibbles in A	0	1	0	0	0	1	1	1	1	1				
	XRL A,Rr	(A)←(A) XOR (Rr)	Logical XOR contents of designated register with A	1	1	0	1	1	r	r	r	1	1				
	XRL A,@Rr	(A)←(A) XOR ((Rr))	Logical XOR indirect the contents of data memory location with A	1	1	0	1	0	0	0	r	1	1				
	XRL A,#data	(A)←(A) XOR data	Logical XOR Immediate data with A	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2				

	mnemonic	function	description	D7	D6	D5	D4	D3	D2	D1	D0	cycles	bytes	C	AC	F0	F1	BS
BRANCH	DJNZ Rr, addr	$(Rr) \leftarrow (Rr) - 1$ if (Rr) not zero: $(PC_{0-7}) \leftarrow addr$	Decrement the specified register and test contents	1 a_7	1 a_6	1 a_5	0 a_4	1 a_3	r a_2	r a_1	r a_0	2	2					
	JBb addr	$(PC_{0-7}) \leftarrow addr$ if $Bb = 1$; $(PC) \leftarrow (PC) + 2$ if $Bb = 0$	Jump to specified address if accumulator bit is set	b_2 a_7	b_1 a_6	b_0 a_5	1 a_4	0 a_3	0 a_2	1 a_1	0 a_0	2	2					
	JC addr	$(PC_{0-7}) \leftarrow addr$ if $C = 1$; $(PC) \leftarrow (PC) + 2$ if $C = 0$	Jump to specified address if carry flag is set	1 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JF0 addr	$(PC_{0-7}) \leftarrow addr$ if $(F0 = 1$; $(PC) \leftarrow (PC) + 2$ if $F0 = 0$	Jump to specified address if flag F0 is set	1 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JF1 addr	$(PC_{0-7}) \leftarrow addr$ if $F1 = 1$; $(PC) \leftarrow (PC) + 2$ if $F1 = 0$	Jump to specified address if flag F1 is set	0 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JMP addr	$(PC_{8-10}) \leftarrow addr_{8-10}$ $(PC_{0-7}) \leftarrow addr_{0-7}$ $(PC_{11}) \leftarrow (DBF)$	Direct jump to specified address within the 2K address block	a_{10} a_7	a_9 a_6	a_8 a_5	0 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2					
	JMPP @A	$(PC_{0-7}) \leftarrow ((A))$	Jump indirect to specified address within address page	1	0	1	1	0	0	1	1	2	1					
	JNC addr	$(PC_{0-7}) \leftarrow addr$ if $C = 0$; $(PC) \leftarrow (PC) + 2$ if $C = 1$	Jump to specified address if carry flag is LOW	1 a_7	1 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNI	$(PC_{0-7}) \leftarrow addr$ if $\overline{INT} = 0$; $(PC) \leftarrow (PC) + 2$ if $\overline{INT} = 1$	Jump to specified address if $\overline{INT}$ input is LOW	1 a_7	0 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNT0 addr	$(PC_{0-7}) \leftarrow addr$ if $T0 = 0$; $(PC) \leftarrow (PC) + 2$ if $T0 = 1$	Jump to specified address if T0 is LOW	0 a_7	0 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNT1 addr	$(PC_{0-7}) \leftarrow addr$ if $T1 = 0$; $(PC) \leftarrow (PC) + 2$ if $T1 = 1$	Jump to specified address if T1 is LOW	0 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					

BRANCH (continued)	JNZ addr	(PC_{0-7})←addr if A ≠ 0; (PC)←(PC) + 2 if A = 0	Jump to specified address if A is non-zero	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2			
	JTF addr	(PC_{0-7})←addr if TF = 1; (PC)←(PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2			
	JT0 addr	(PC_{0-7})←addr if T0 = 1; (PC)←(PC) + 2 if T0 = 0	Jump to specified address if T0 = 1	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2			
	JT1 addr	(PC_{0-7})←addr if T1 = 1; (PC)←(PC) + 2 if T1 = 0	Jump to specified address if T1 = 1	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2			
	JZ addr	(PC_{0-7})←addr if A = 0; (PC)←(PC) + 2 if A ≠ 0	Jump to specified address if A is zero	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2			
CONTROL	EN I		Enable external ($\overline{INT}$) interrupt	0	0	0	0	0	1	0	1	1	1			
	DIS I		Disable external ($\overline{INT}$) interrupt	0	0	0	1	0	1	0	1	1	1			
	SEL RB0	(BS)←0	Select bank 0 (locations 0–7) of data memory	1	1	0	0	0	1	0	1	1	1			●
	SEL RB1	(BS)←1	Select bank 1 (locations 24–31) of data memory	1	1	0	1	0	1	0	1	1	1			●
	SEL MB0	(DBF)←0	Select program memory bank 0; addresses 0–2047	1	1	1	0	0	1	0	1	1	1			
	SEL MB1	(DBF)←1	Select program memory bank 1; addresses 2048–4095	1	1	1	1	0	1	0	1	1	1			
	ENT0 CLK		Enable clock output onto T0	0	1	1	1	0	1	0	1	1	1			

	mnemonic	function	description	instruction code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	cycles	bytes	flags C	AC	F0	F1	BS
DATA MOVES	MOV A,#data	(A)←data	Move immediate data into A	0 d_7	0 d_6	1 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
	MOV A,Rr	(A)←(Rr) for r = 0–7	Move the contents of the designated register into A	1	1	1	1	1	r	r	r	1	1					
	MOV A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of data memory into A	1	1	1	1	0	0	0	r	1	1					
	MOV A,PSW	(A)←(PSW)	Move contents of the program status word into A	1	1	0	0	0	1	1	1	1	1					
	MOV Rr,#data	(Rr)←data for r = 0–7	Move immediate data into the designated register	1 d_7	0 d_6	1 d_5	1 d_4	1 d_3	r d_2	r d_1	r d_0	2	2					
	MOV Rr,A	(Rr)←(A) for r = 0–7	Move A contents into the designated register	1	0	1	0	1	r	r	r	1	1					
	MOV @Rr,A	((Rr))←(A) for r = 0–1	Move indirect A contents into data memory location	1	0	1	0	0	0	0	r	1	1					
	MOV @Rr,#data	((Rr))←data for r = 0–1	Move indirect the specified data into data memory	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	r d_0	2	2					
	MOV PSW,A	(PSW)←A	Move contents of A into the program status word	1	1	0	1	0	1	1	1	1	1	●	●	●		●
	MOVP A,@A	(A)←((A))	Move data in the current page into A	1	0	1	0	0	0	1	1	2	1					
	MOVP3 A,@A	(A)←((A)) in page 3	Move data in page 3 into A	1	1	1	0	0	0	1	1	2	1					
	MOVX A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of external memory location into A	1	0	0	0	0	0	0	r	2	1					
	MOVX @Rr,A	((Rr))←(A) for r = 0–1	Move indirect the contents of A into external memory	1	0	0	1	0	0	0	r	2	1					
	XCH A,Rr	(A)↔(Rr) for r = 0–7	Exchange A with designated register contents	0	0	1	0	1	r	r	r	1	1					
	XCH A,@Rr	(A)↔((R)) for r = 0–1	Exchange indirect A contents with location in data memory	0	0	1	0	0	0	0	r	1	1					

D.M.	XCHD A,@Rr	$(A_{0-3}) \leftrightarrow (Rr_{0-3})$ for r = 0–1	Exchange indirect 4-bit contents of A with data memory	0	0	1	1	0	0	0	r	1	1				
FLAGS	CPL C	(C)←NOT (C)	Complement content of carry bit	1	0	1	0	0	1	1	1	1	1	●			
	CPL F0	(F0)←NOT (F0)	Complement content of flag F0	1	0	0	1	0	1	0	1	1	1			●	
	CPL F1	(F1)←NOT (F1)	Complement content of flag F1	1	0	1	1	0	1	0	1	1	1				●
	CLR C	(C)←0	Clear content of carry bit to 0	1	0	0	1	0	1	1	1	1	1	●			
	CLR F0	(F0)←0	Clear content of flag F0 to 0	1	0	0	0	0	1	0	1	1	1			●	
	CLR F1	(F1)←0	Clear content of flag F1 to 0	1	0	1	0	0	1	0	1	1	1				●
INPUT/OUTPUT	ANL BUS,#data	(BUS)←(BUS) AND data	Logical AND immediate data with BUS	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2				
	ANL Pp,#data	(Pp)←(Pp) AND data; p = 1–2	Logical AND immediate data with designated port (1 or 2)	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2				
	ANLD Pp,A	(Pp)←(Pp) AND (A_{0-3}); p = 4–7	Logical AND contents of A with designated port (4–7)	1	0	0	1	1	1	p	p	2	1				
	IN A,Pp	(A)←(Pp) p = 1–2	Input data from designated port (1–2) into A	0	0	0	0	1	0	p	p	2	1				
	INS A,BUS	(A)←(BUS)	Input strobed BUS data into A	0	0	0	0	1	0	0	0	1	2				
	MOVD A,Pp	(A_{0-3})←(Pp); p = 4–7 (A_{4-7})←0	Move contents of designated port (4–7) into A	0	0	0	0	1	1	p	p	2	1				
	MOVD Pp,A	(Pp)←(A_{0-3}) p = 4–7	Move contents of A to designated port (4–7)	0	0	1	1	1	1	p	p	1	1				
	ORLD Pp,A	(Pp)←(Pp) OR (A_{0-3}); p = 4–7	Logical OR contents of A with designated port (4–7)	1	0	0	0	1	1	p	p	1	1				
	ORL BUS,#data	(BUS)←(BUS) OR data	Logical OR immediate data with BUS	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2				
	ORL Pp,#data	(Pp)←(Pp) OR data p = 1–2	Logical OR immediate data with designated port (1–2)	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2				
	OUTL BUS,A	(BUS)←(A)	Output contents A onto BUS	0	0	0	0	0	0	1	0	1	2				
	OUTL Pp,A	(Pp)←(A) p = 1–2	Output contents A to designated port (1–2)	0	0	1	1	1	0	p	p	1	1				

	mnemonic	function	description	instruction code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	cycles	bytes	flags C	AC	F0	F1	BS
REGISTER	DEC Rr	(Rr)←(Rr) −1 for r = 0–7	Decrement contents of designated register by 1	1	1	0	0	1	r	r	r	1	1					
	INC Rr	(Rr)←(Rr) + 1 for r = 0–7	Increment contents of designated register by 1	0	0	0	1	1	r	r	r	1	1					
	INC @Rr	((R))←((Rr)) + 1 for r = 0–1	Increment indirect the contents of data memory location by 1	0	0	0	1	0	0	0	r	1	1					
SUBROUTINE	CALL addr	((SP))←(PC), (PSW_{4-7}) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$	Call designated subroutine	a_{10} a_7	a_9 a_6	a_8 a_5	1 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2					
	RET	(SP)←(SP) −1 (PC)←((SP))	Return from subroutine without restoring program status word	1	0	0	0	0	0	1	1	2	1					
	RETR	(SP)←(SP) −1 (PC)←((SP)) (PSW_{4-7})←((SP))	Return from subroutine restoring program status word	1	0	0	1	0	0	1	1	2	1					
TIMER/COUNTER	EN TCNTI		Enable timer/counter interrupt	0	0	1	0	0	1	0	1	1	1					
	DIS TCNTI		Disable timer/counter interrupt	0	0	1	1	0	1	0	1	1	1					
	MOV A,T	(A)←(T)	Move contents of timer/counter into A	0	1	0	0	0	0	1	0	1	1					
	MOV T,A	(T)←(A)	Move contents of A into timer/counter	0	1	1	0	0	0	1	0	1	1					
	STOP TCNT		Stop count for event counter or timer	0	1	1	0	0	1	0	1	1	1					
	STRT CNT		Start count for event counter	0	1	0	0	0	1	0	1	1	1					
	STRT T		Start count for timer	0	1	0	1	0	1	0	1	1	1					
	NOP		No operation	0	0	0	0	0	0	0	0	1	1					

Table 2 Instruction timing (see also Figs 11 and 12)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	fetch instruction	increment program counter	–	increment timer	–	–	read port	• –	–	–
OUTL P,A			–		output to port	–	–	• –	–	–
ANL P,#data		•	–		read port	fetch immediate data	–	•increment program counter	output to port	–
ORL P,#data		•	–		read port	fetch immediate data	–	•increment program counter	output to port	–
INS A,BUS			–		–	–	read port	• –	–	–
OUTL BUS,A			–		output to port	–	–	• –	–	–
ANL BUS,#data		•	–		read port	fetch immediate data	–	•increment program counter	output to port	–
ORL BUS,#data		•	–		read port	fetch immediate data	–	•increment program counter	output to port	–
MOVX @R,A			output RAM address		output data to RAM	–	–	• –	–	–
MOVX A,@R			output RAM address		–	–	read data	• –	–	–
MOVD A,P	fetch instruction	increment program counter	output opcode/ address	increment timer	–	–	read P2 lower	• –	–	–

Continued on next page.

Table 2 Instruction timing (continued)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
MOVD P,A	fetch instruction	increment program counter	output opcode/ address	increment timer	output data to P2 lower	–	–	• –	–	–
ANLD P,A			output opcode/ address		output data	–	–	• –	–	–
ORLD P,A			output opcode/ address		output data	–	–	• –	–	–
J (conditional)		•	sample condition	increment timer	–	fetch immediate data	–	•update program counter	–	–
STRT CNT STRT T		•	–	–	start counter					
STOP TCNT		•	–	–	stop counter					
EN I		•	–	enable interrupt	–					
DIS I		•	–	disable interrupt	–					
ENT0 CLK	fetch instruction	•increment program counter	–	enable clock	–					

- Valid instruction addresses are output at this time if external program memory is being accessed.

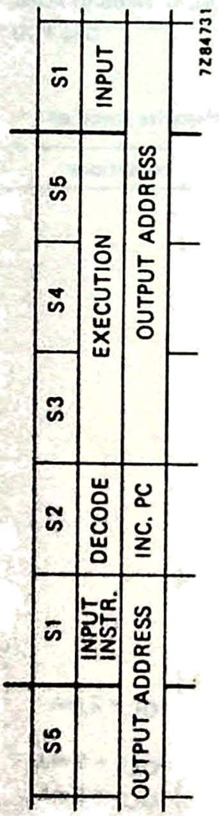

S5
S1
S2
S3
S4
S5
S1
INPUT INSTR.
DECODE
EXECUTION
INPUT
OUTPUT ADDRESS
INC. PC
OUTPUT ADDRESS
7Z84731

Fig. 11 Instruction cycle.

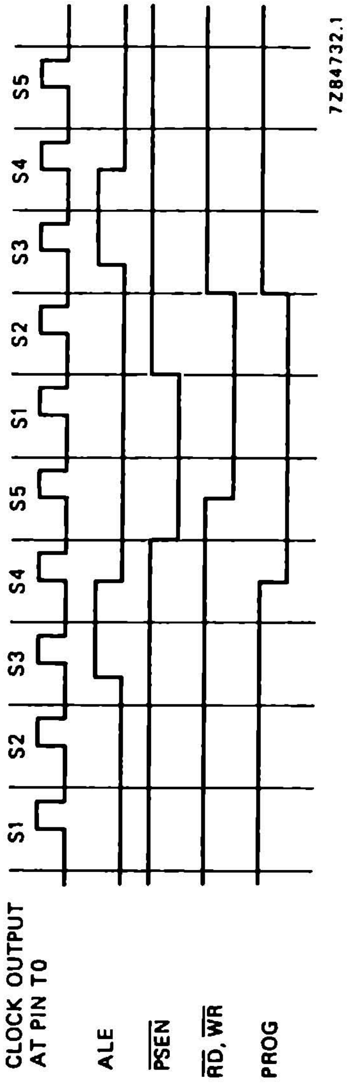

CLOCK OUTPUT AT PIN T0
ALE
PSEN
RD, WR
PROG
S1
S2
S3
S4
S5
S1
S2
S3
S4
S5
7Z84732.1

Fig. 12 Instruction cycle timing.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage with respect to V_{SS}				
except input EA	V_I		−0,5 to + 7	V
input EA	V_I		−0,5 to + 15	V
D.C. current into any input or output	$\pm I_I, \pm I_O$	max.	10	mA
Total power dissipation	P_{tot}	max.	1,5	W
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating ambient temperature range	T_{amb}		0 to + 70	°C

D.C. CHARACTERISTICS

$V_{SS} = 0$ V; $T_{amb} = 0$ to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit	conditions
Supply voltage	$V_{CC} = V_{DD}$	4,5	5,0	5,5	V	
Supply current	I_{DD}	–	–	9	mA	
Supply current (total)	$I_{CC} + I_{DD}$	–	–	90	mA	
Inputs						
Input voltage LOW all inputs except XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IL}	−0,5	–	0,8	V	
Input voltage LOW XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IL1}	−0,5	–	0,6	V	
Input voltage HIGH all inputs except XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IH}	2,0	–	V_{CC}	V	
Input voltage HIGH XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IH1}	3,8	–	V_{CC}	V	
Outputs						
Output voltage LOW; BUS	V_{OL}	–	–	0,45	V	$I_{OL} = 2$ mA
Output voltage LOW $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE	V_{OL1}	–	–	0,45	V	$I_{OL1} = 2$ mA
Output voltage LOW; PROG	V_{OL2}	–	–	0,45	V	$I_{OL2} = 1$ mA
Output voltage LOW all other outputs	V_{OL3}	–	–	0,45	V	$I_{OL3} = 1,6$ mA
Output voltage HIGH; BUS	V_{OH}	2,4	–	–	V	$-I_{OH} = 100$ μA
Output voltage HIGH $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE	V_{OH1}	2,4	–	–	V	$-I_{OH1} = 100$ μA
Output voltage HIGH all other outputs	V_{OH2}	2,4	–	–		$-I_{OH2} = 50$ μA

parameter	symbol	min.	typ.	max.	unit	conditions
Input leakage current T1, $\overline{INT}$	$\pm I_{IL}$	–	–	10	μA	$V_{SS} < V_I < V_{CC}$
Input leakage current P10 to P17, P20 to P27, EA, $\overline{SS}$	$-I_{IL1}$	–	–	500	μA	$V_{SS} + 0{,}45\ V < V_I < V_{CC}$
Output leakage current BUS, T0 (high impedance)	$\pm I_{OL}$	–	–	10	μA	$V_{SS} + 0{,}45\ V < V_I < V_{CC}$

A.C. CHARACTERISTICS

$V_{CC} = V_{DD} = 5\ \text{V} \pm 10\%$; $V_{SS} = 0\ \text{V}$; $T_{amb} = 0$ to $+ 70\ ^{o}\text{C}$

See waveforms Figs 13, 14 and 15

parameter	symbol	f = 6 MHz min.	f = 6 MHz max.	f = 11 MHz min.	f = 11 MHz max.	unit	conditions note 1
ALE pulse width	t_{LL}	400	–	150	–	ns	
Address set-up time to ALE	t_{AL}	150	–	70	–	ns	
Address hold time from ALE	t_{LA}	80	–	50	–	ns	
Control pulse width $\overline{\text{PSEN}}$, $\overline{\text{RG}}$, $\overline{\text{WR}}$	t_{CC}	700	–	300	–	ns	
Data set-up time before $\overline{\text{WR}}$	t_{DW}	500	–	250	–	ns	
Data hold time after $\overline{\text{WR}}$	t_{WD}	120	–	40	–	ns	note 2
Cycle time	t_{CY}	2,5	15.0	1,36	15.0	μs	
Data hold time	t_{DR}	0	200	0	100	ns	
$\overline{\text{PSEN}}$, $\overline{\text{RD}}$ to data input	t_{RD}	–	500	–	200	ns	
Address set-up time to $\overline{\text{WR}}$	t_{AW}	230	–	200	–	ns	
Address set-up time to data input	t_{AD}	–	950	–	400	ns	
Address floating to $\overline{\text{RD}}$, $\overline{\text{PSEN}}$	t_{AFC}	0	–	–10	–	ns	

Notes

1. Control outputs: $C_L = 80$ pF
   Bus outputs: $C_L = 150$ pF
2. Bus high-impedance load: 20 pF

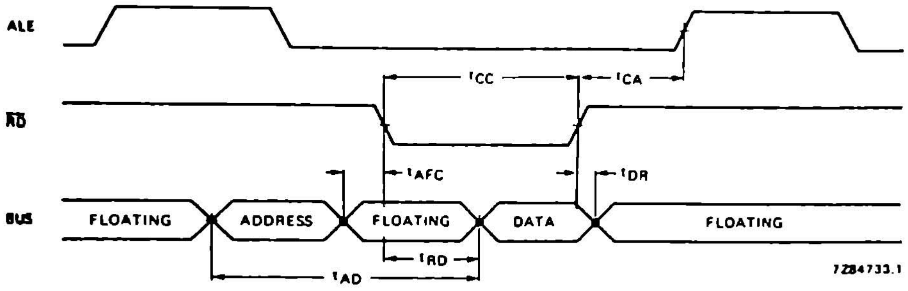

ALE
RD
BUS
tCC
tCA
tAFC
tDR
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
tRD
tAD
7Z84733.1

Fig. 13 Read from external data memory.

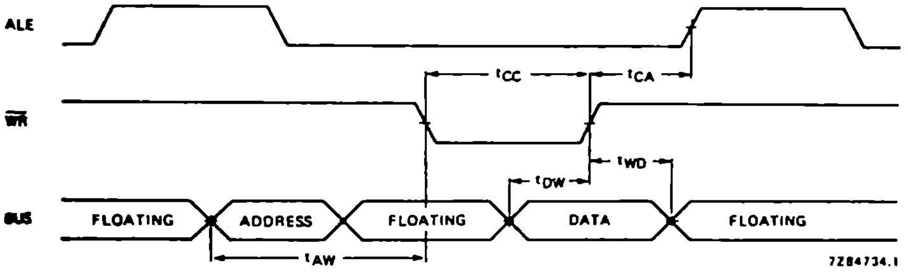

ALE
WR
BUS
tCC
tCA
tWD
tDW
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
tAW
7Z84734.1

Fig. 14 Write to external data memory.

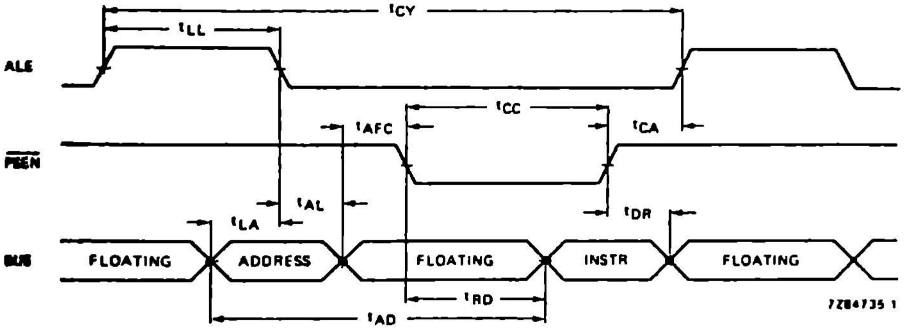

tCY
tLL
ALE
tAFC
tCC
tCA
PSEN
tAL
tLA
tDR
BUS
FLOATING
ADDRESS
FLOATING
INSTR
FLOATING
tRD
tAD
7Z84735.1

Fig. 15 Instruction fetch from external program memory.

A.C. CHARACTERISTICS

Port 2 timing (see Fig. 16)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C

parameter	symbol	f = 6 MHz min.	f = 6 MHz max.	f = 11 MHz min.	f = 11 MHz max.	unit
Port control set-up time before falling edge of PROG	t_{CP}	110	–	60	–	ns
Port control hold time after falling edge of PROG	t_{PC}	100	–	55	–	ns
PROG to time P_2 input must be valid	t_{PR}	–	810	–	440	ns
Input data hold time	t_{PF}	0	150	0	80	ns
Output data set-up time	t_{DP}	250	–	130	–	ns
Output data hold time	t_{PD}	65	–	35	–	ns
PROG pulse width	t_{PP}	1200	–	650	–	ns
Port 2 I/O data set-up time	t_{PL}	350	–	200	–	ns
Port 2 I/O data hold time	t_{LP}	150	–	80	–	ns

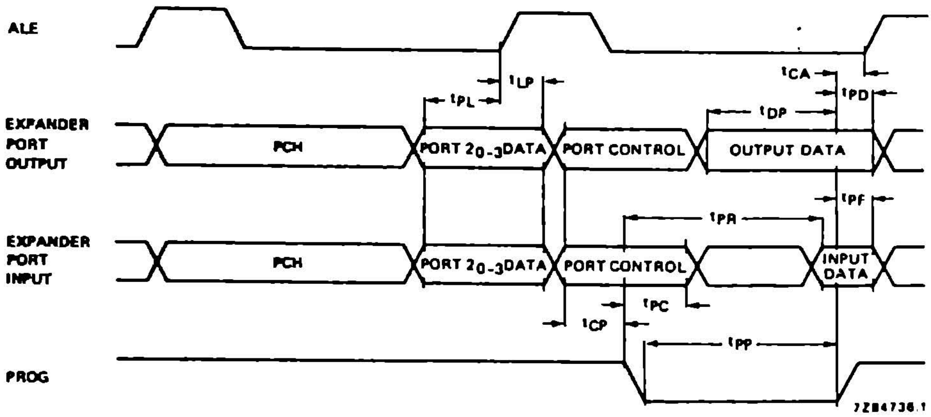
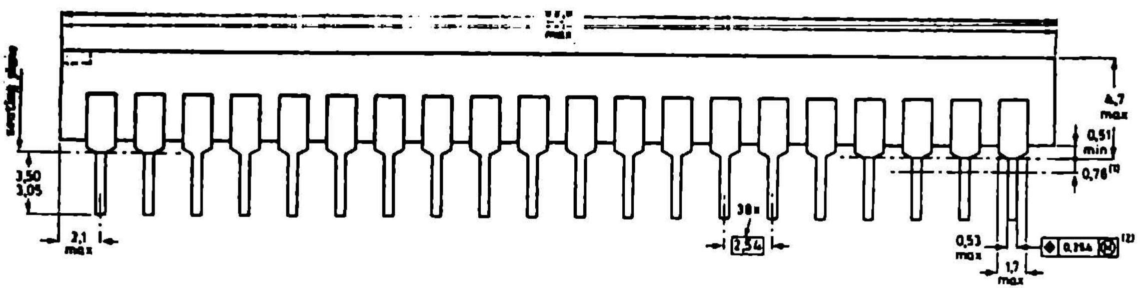
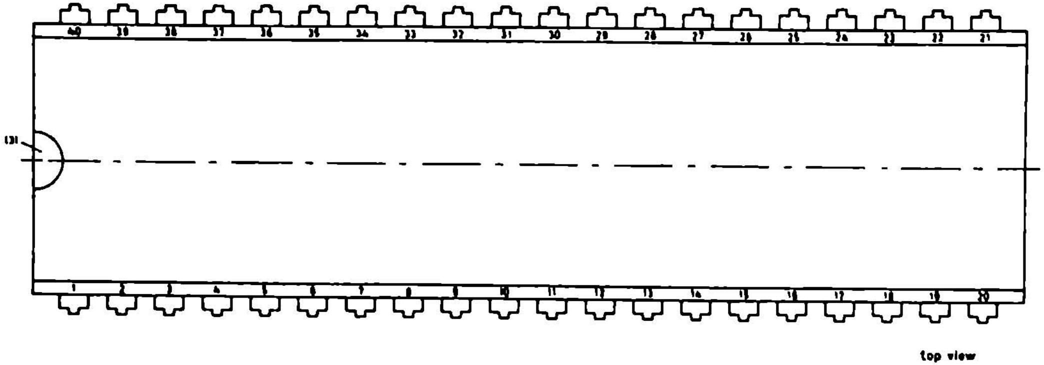
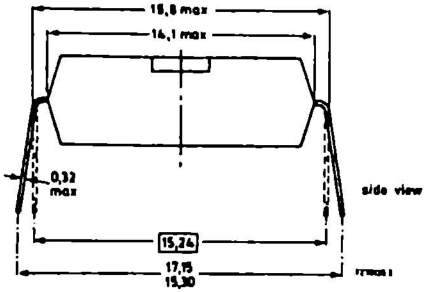

ALE
EXPANDER PORT OUTPUT
PCH
PORT 2₀-₃DATA
PORT CONTROL
OUTPUT DATA
EXPANDER PORT INPUT
PCH
PORT 2₀-₃DATA
PORT CONTROL
INPUT DATA
PROG
tLP
tPL
tCA
tPD
tDP
tPF
tPR
tPC
tCP
tPP
7Z84736.1

Fig. 16 Port 2 timing.

40-LEAD DUAL IN-LINE; PLASTIC (SOT-129)


4,7
max
0,51
min
0,78(1)
0,254 (M)(2)
0,53
max
1,7
max
38×
2,54
3,50
3,05
2,1
max
seating plane




15,8 max
16,1 max
0,32
max
15,24
17,15
15,90
side view

Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

SOLDERING

See next page.

SOLDERING

1. By hand

Apply the soldering iron below the seating plane (or not more than 2 mm above it).
If its temperature is below 300 °C it must not be in contact for more than 10 seconds; if between 300 °C and 400 °C, for not more than 5 seconds.

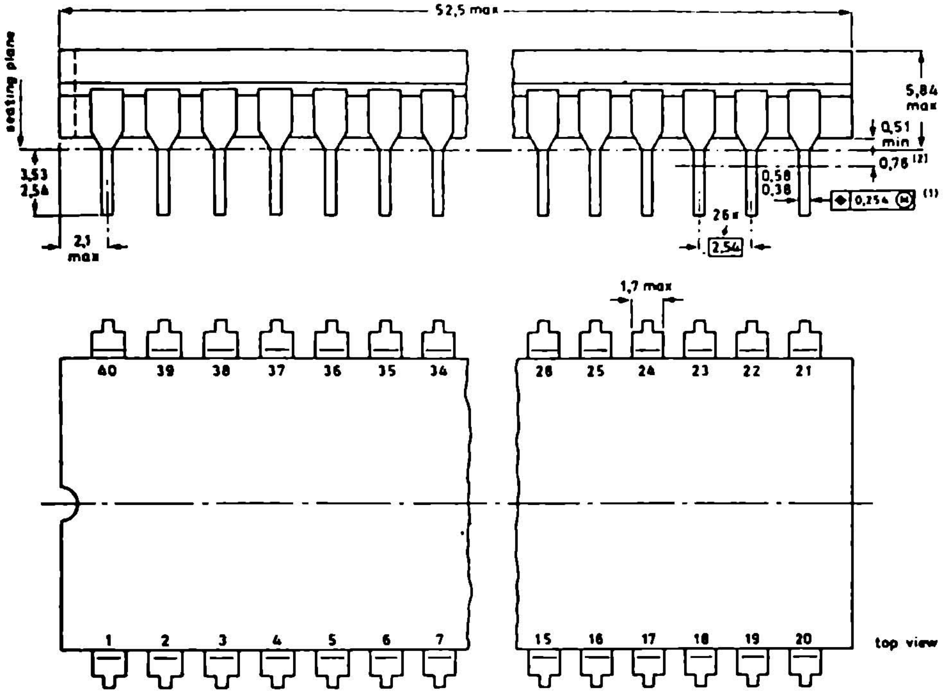
2. By dip or wave

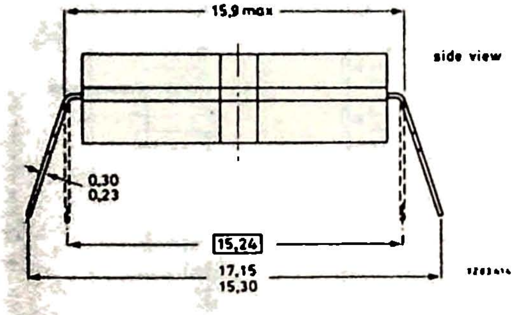
The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.
The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

3. Repairing soldered joints

The same precautions and limits apply as in (1) above.

40-LEAD DUAL IN-LINE; CERAMIC (SOT-145)


52,5 max
seating plane
5,84 max
0,51 min
0,76 (2)
0,58
0,38
0,254 M (1)
3,53
2,54
2,1 max
26 x
2,54
1,7 max
40 39 38 37 36 35 34
28 25 24 23 22 21
1 2 3 4 5 6 7
15 16 17 18 19 20
top view


15,9 max
side view
0,30
0,23
15,24
17,15
15,30

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

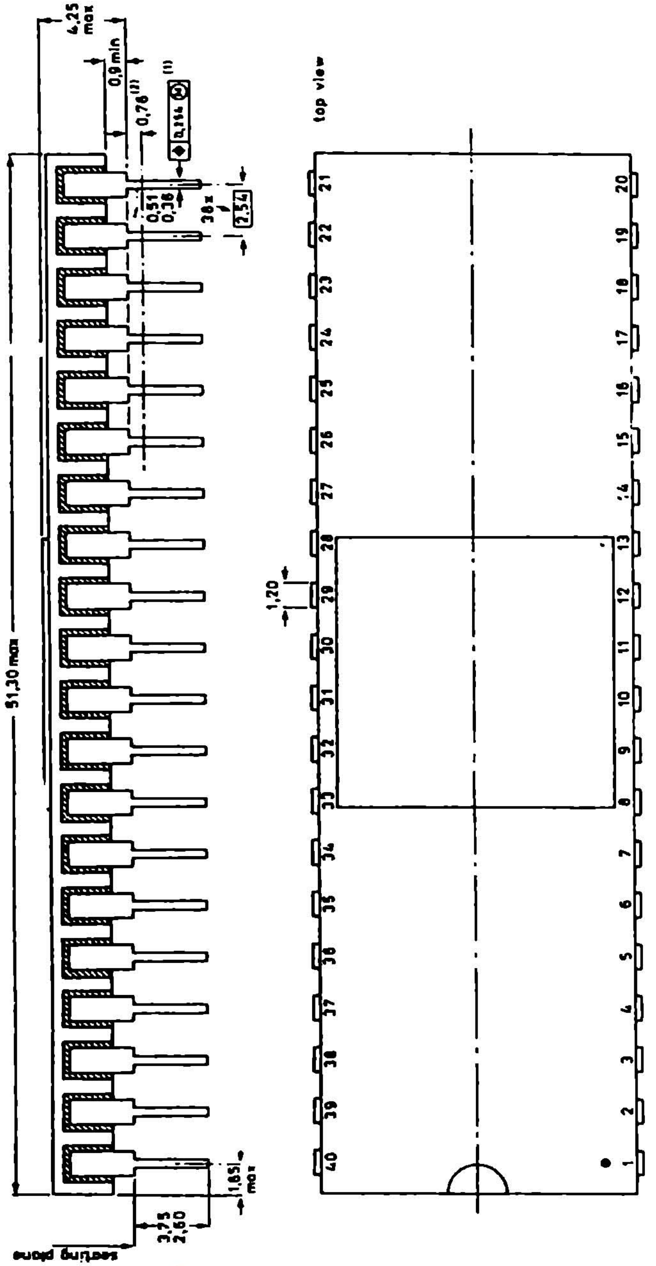
(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

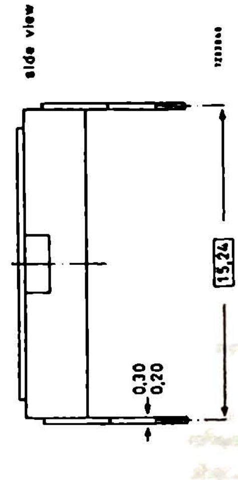
Remarks

1. Leads are given positive misalignment so that they grip after insertion.
2. Leads are Ni-Fe, pure tin plated.

40-LEAD DUAL IN-LINE; METAL-CERAMIC (SOT-88B)



4,25 max
0,9 min
0,76 (2)
0,254 Ⓜ (1)
0,51
0,38
38x
2,54
51,30 max
1,65 max
3,75
2,60
seating plane
top view
1,20
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40



side view
0,30
0,20
15,24

⊕ Positional accuracy.

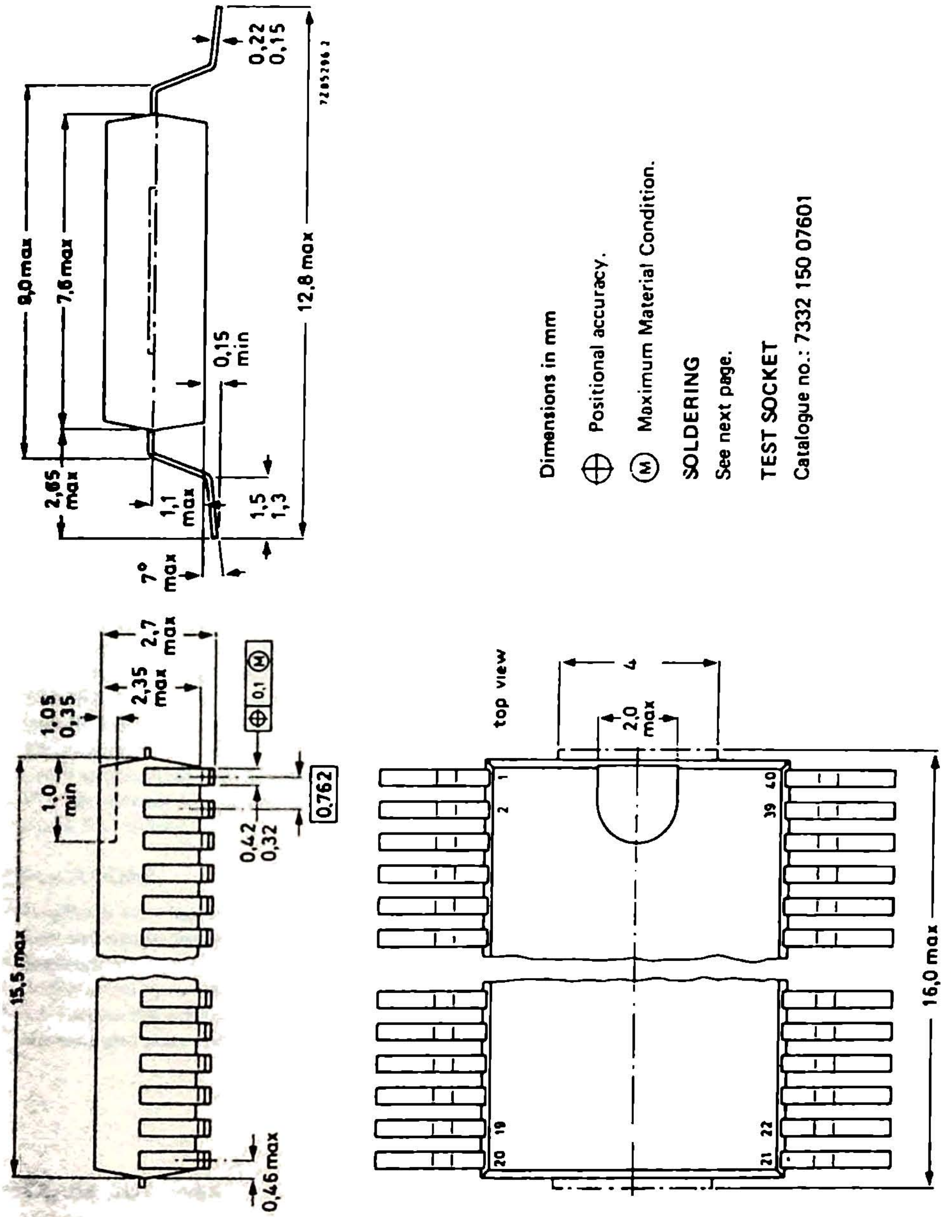
Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

40-LEAD FLAT PACK; PLASTIC (VSO-40; SOT-158)


0,22
0,15
7Z85296.2
9,0 max
7,6 max
12,8 max
0,15
min
2,65
max
1,1
max
1,5
1,3
7°
max
2,7
max
2,35
max
1,05
0,35
1,0
min
0,1 M
0,762
0,42
0,32
15,5 max
0,46 max
top view
4
2,0
max
1
2
40
39
20
19
21
22
16,0 max

Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

SOLDERING

See next page.

TEST SOCKET

Catalogue no.: 7332 150 07601

SOLDERING

1. Soldering iron or pulse heated solder tool

Apply the heating tool to the flat part of the pin only.
Limit the contact time to maximum 10 seconds up to 300 °C, or 5 seconds up to maximum 400 °C.
When using the proper tools, up to 20 pins (at one side of the device) can be soldered in one operation with 2 to 5 seconds and 270 to 320 °C.

2. By dip or wave

The maximum permissible temperature of the solder is 260 °C. The permissible total time of immersing the whole package in the bath is 10 seconds, if it is allowed to cool down to less than 150 °C within 6 seconds.

3. Repairing soldered joints

The same precautions and limits apply as in (1) above.
If the vertical part of the pin needs heating, reduce the soldering iron temperature to 260 °C.

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The MAB8050H family of single-chip 8-bit microcomputers are fabricated in H-MOS.
Two interchangeable (pin compatible) versions are available:

- The MAB8050H with resident mask-programmed ROM,
- The MAB8040H without resident program memory for use with external EPROM/ROM.

The MAB8050H family are designed to be efficient control processors as well as arithmetic processors. Their instruction set allows the user to directly set and reset individual I/O lines as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions enable efficient implementation of standard logic functions. Code efficiency is high; over 70% of the instructions are single byte; all others are two byte.

An on-chip 8-bit counter is provided, which can count either machine cycles (÷ 32) or external events. The counter can be programmed to cause an interrupt to the processor.
Program and data memories can be expanded using standard devices. Input/output capabilities can be expanded using standard devices.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 40-pin package
- 4K x 8 ROM, 256x8 RAM, 27 I/O lines
- Internal counter/timer
- Internal oscillator, clock driver
- Single-level interrupts: external and counter/timer
- 17 internal registers: accumulator, 16 addressable registers
- Over 90 instructions: 70% single byte
- All instructions: 1 or 2 cycles (1,36/2,5 μs per cycle)
- Easily expandable memory and I/O
- TTL compatible inputs and outputs
- Single 5 V supply

APPLICATIONS

- Peripheral interfaces and controllers
- Test and measurement instruments
- Sequencers
- Audio/video systems
- Environmental control systems
- Modems and data enciphering

PACKAGE OUTLINES

MAB8050HP: 40-lead DIL; plastic (SOT-129).
MAB8050HD: 40-lead DIL; ceramic (SOT-145).
MAB8050HD: 40-lead DIL; metal-ceramic (SOT-88B).

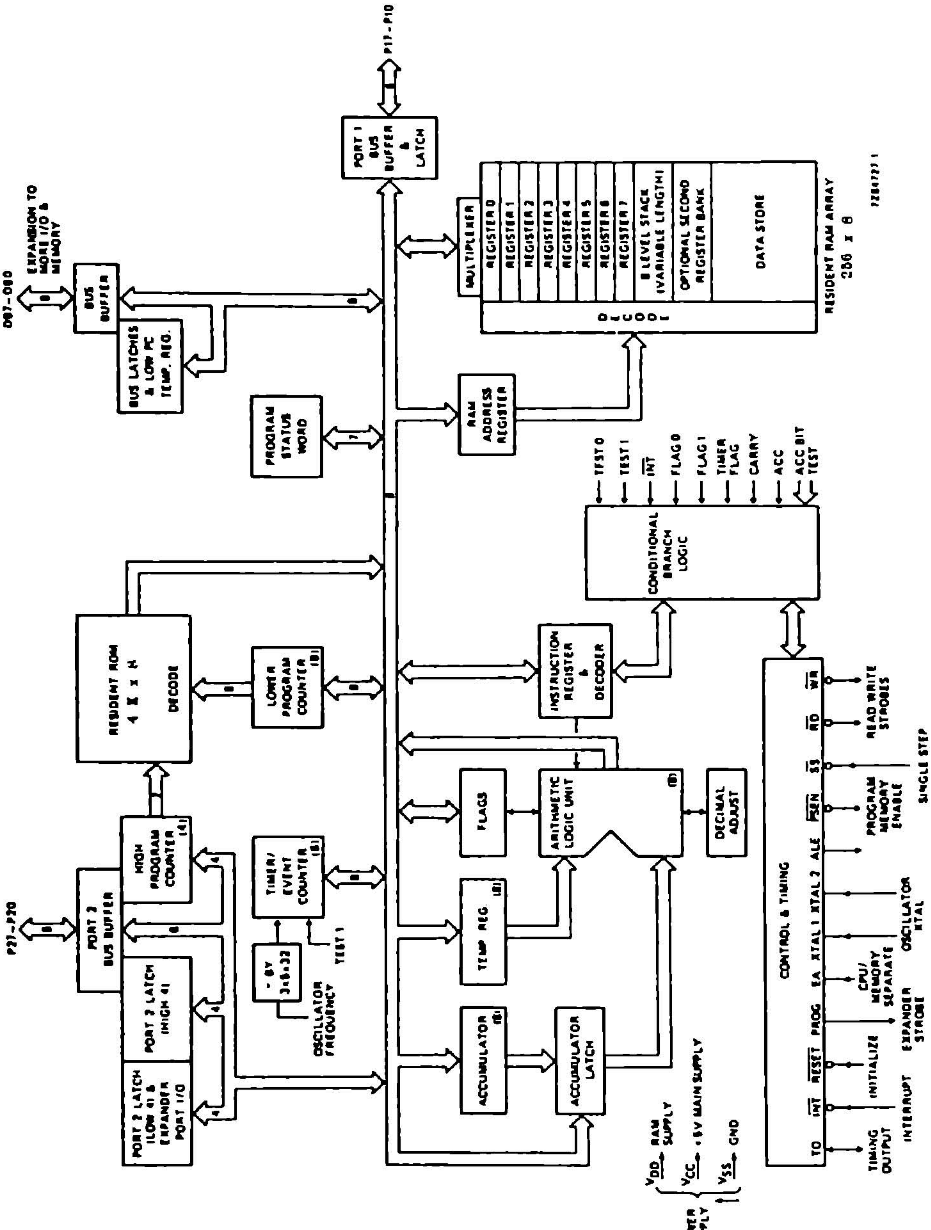

P17 - P10
PORT 1 BUS BUFFER & LATCH
D07 - D00
EXPANSION TO MORE I/O & MEMORY
BUS BUFFER
BUS LATCHES & LOW PC TEMP. REG.
8
MULTIPLEXER
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
8 LEVEL STACK (VARIABLE LENGTH)
OPTIONAL SECOND REGISTER BANK
DATA STORE
DECODE
RESIDENT RAM ARRAY 256 x 8
7Z84772.1
PROGRAM STATUS WORD
RAM ADDRESS REGISTER
TEST 0
TEST 1
INT
FLAG 0
FLAG 1
TIMER FLAG
CARRY
ACC
ACC BIT TEST
CONDITIONAL BRANCH LOGIC
RESIDENT ROM 4 K x 8
DECODE
LOWER PROGRAM COUNTER (8)
INSTRUCTION REGISTER & DECODER
P27 - P20
PORT 2 BUS BUFFER
HIGH PROGRAM COUNTER (4)
PORT 2 LATCH (HIGH 4)
PORT 2 LATCH (LOW 4) & EXPANDER PORT I/O
4
TIMER/ EVENT COUNTER (8)
÷ 15 (3x5x32)
TEST 1
OSCILLATOR FREQUENCY
FLAGS
ARITHMETIC LOGIC UNIT (8)
DECIMAL ADJUST
TEMP REG. (8)
ACCUMULATOR (8)
ACCUMULATOR LATCH
VDD RAM SUPPLY
VCC +5 V MAIN SUPPLY
VSS GND
POWER SUPPLY
CONTROL & TIMING
T0
INT
RESET
PROG
EA
XTAL 1
XTAL 2
ALE
PSEN
SS
RD
WR
TIMING OUTPUT
INTERRUPT
INITIALIZE
EXPANDER STROBE
CPU/ MEMORY SEPARATE
OSCILLATOR XTAL
PROGRAM MEMORY ENABLE
SINGLE STEP
READ WRITE STROBES

Fig. 1 Block diagram.

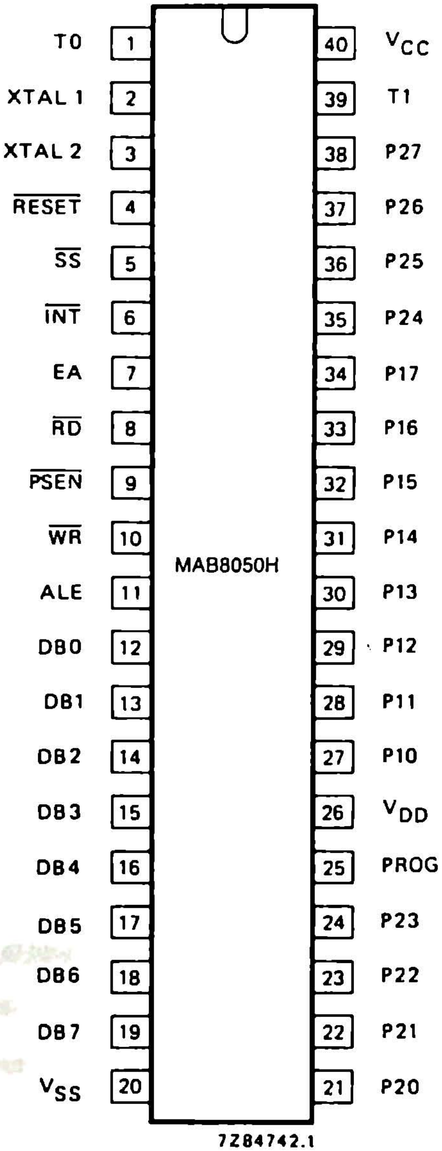

T0
XTAL 1
XTAL 2
$\overline{\text{RESET}}$
$\overline{\text{SS}}$
$\overline{\text{INT}}$
EA
$\overline{\text{RD}}$
$\overline{\text{PSEN}}$
$\overline{\text{WR}}$
ALE
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
V_{SS}
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
MAB8050H
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
V_{CC}
T1
P27
P26
P25
P24
P17
P16
P15
P14
P13
P12
P11
P10
V_{DD}
PROG
P23
P22
P21
P20
7Z84742.1

Fig. 2 Pinning diagram; for pin designation see next page.

PIN DESIGNATION

designation	pin no.	function
DB0–DB7	12–19	BUS. Bidirectional I/O port that can be read or written using the $\overline{RD}$ and $\overline{WR}$ strobes. This port can also be statically latched. Contains the 8 lower order address bits during external memory access and receives the addressed instruction under control of $\overline{PSEN}$. $\overline{PSEN}$, ALE, $\overline{RD}$ and $\overline{WR}$ determine whether the access is an instruction fetch or a read/write access to external RAM.
P10–P17	27–34	**Port 1.** 8-bit quasi-bidirectional I/O port (note 1).
P20–P27	21–24, 35–38	**Port 2.** 8-bit quasi-bidirectional I/O port (note 1). P20–P23 contain the 4 higher order address bits during an access of external program memory and also serve as a 4-bit I/O expander bus for the 8243.
PROG	25	**Output strobe** (active LOW) for 8243 I/O expander.
T0	1	**Input pin** sensed using the JT0 and JNT0 instructions. **Clock output pin** when designated as such by the ENT0 CLK instruction.
T1	39	**Input pin** sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
$\overline{INT}$	6	**Interrupt input pin.** When LOW causes an interrupt in the current program if external interrupt is enabled. Can also be used as an input, testable using the JNI instruction. Interrupt is disabled during and after a RESET.
$\overline{RESET}$	4	**Reset input pin** used to initialize the microcomputer. Active LOW. During program verification the address is latched by a '0' to '1' transition on $\overline{RESET}$ and the data at the addressed location is output on BUS (note 2).
ALE	11	**Address latch enable.** Occurs each clock cycle and is useful for timing and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0 to DB7 outputs.
$\overline{RD}$	8	**Read** BUS. Active LOW strobe used to gate data onto BUS lines when reading from an external source.
$\overline{WR}$	10	**Write** BUS. Active LOW strobe used to write data from BUS lines to an external designation.
EA	7	**External access** input. When HIGH, forces instruction fetch from external memory.
$\overline{PSEN}$	9	**Program store enable.** Active LOW strobe that occurs only during a fetch from external program memory.
$\overline{SS}$	5	**Single step input.** Active LOW which is used with ALE to cause the microcomputer to execute a single instruction.
XTAL1	2	**One side of crystal** (or inductor) input for internal oscillator. Can also be used as an input for an external timing source (note 2).

designation	pin no.	function
XTAL2	3	Other side of crystal.
V_{SS}	20	Ground.
V_{CC}	40	Power supply, + 5 V.
V_{DD}	26	RAM standby power supply, low power pin.

Notes

1. Each port line can be designated as an input or an output. A line is designated as an input by first writing a logic '1' to the line. $\overline{\text{RESET}}$ sets all lines to logic '1'.
2. Non-standard TTL V_{IH}.

FUNCTIONAL DESCRIPTION

Detailed information available on request.

Program memory

The resident program memory consists of a 4096 byte ROM (MAB8050H); the MAB8040H has no resident program memory.
The total addressing capability is 4096 bytes.
The program memory address space is divided into two 2048-bytes banks MB0 and MB1 (Fig. 3).
Further, the program memory is divided into pages of 256 bytes each. This latter division applies only for conditional branches.
There are three locations in program memory of special importance. These locations contain the first instruction to be executed after one of three events.
The three locations and their contents are:

location 0 – activation, then deactivation of the $\overline{\text{RESET}}$ line.
location 3 – activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled,
location 7 – an overflow of the timer/counter if the T/C interrupt is enabled.

Data memory

The resident data memory, as shown in Fig. 4, consists of a 256 byte RAM. All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1. The first eight locations of RAM (0 to 7) are designated as working registers and are directly addressable by several instructions.
By selecting register bank 1, RAM locations 24 to 31 become the working registers, replacing those in register bank 0 (0 to 7).
RAM locations 8 to 23 are designated as the stack. Two locations (bytes) are used per CALL, allowing up to eight levels of subroutine nesting.
If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM (greater than 384 bytes total) is required, an I/O port can be used to select one (256 byte) bank of external memory at a time.

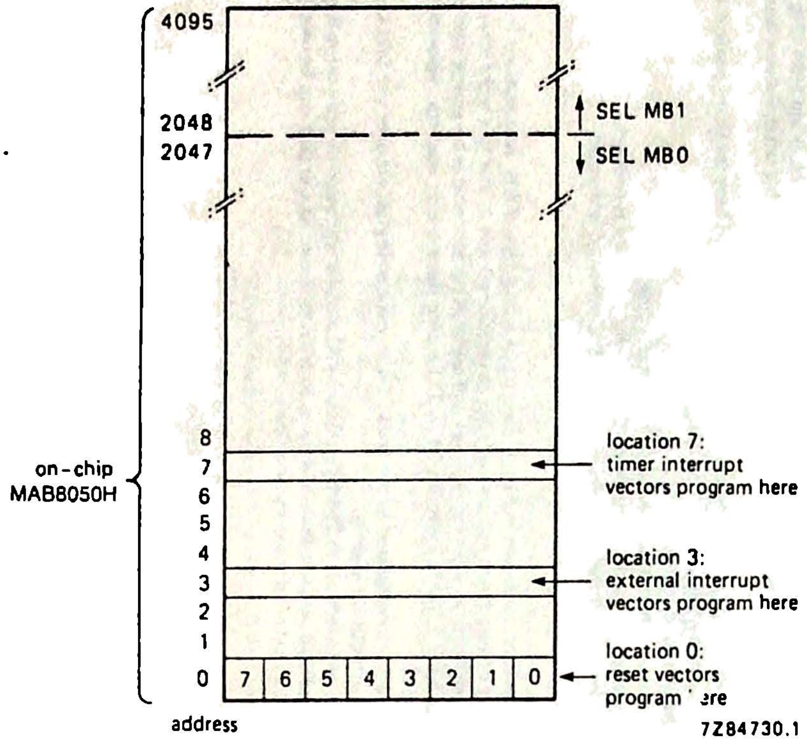

4095
2048
2047
SEL MB1
SEL MB0
8
7
6
5
4
3
2
1
0
7 6 5 4 3 2 1 0
address
on-chip
MAB8050H
location 7:
timer interrupt
vectors program here
location 3:
external interrupt
vectors program here
location 0:
reset vectors
program ' ɜre
7Z84730.1

Fig. 3 Program memory map.

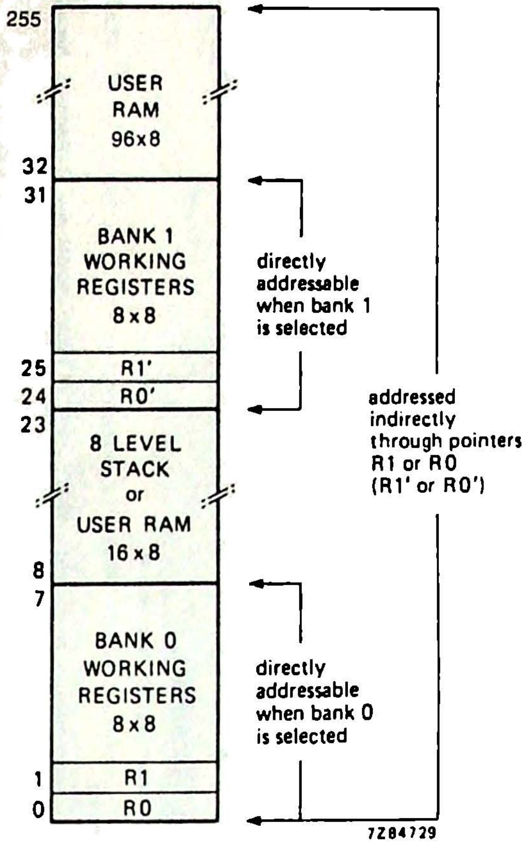

255
USER
RAM
96x8
32
31
BANK 1
WORKING
REGISTERS
8 x 8
25
R1'
24
R0'
23
8 LEVEL
STACK
or
USER RAM
16 x 8
8
7
BANK 0
WORKING
REGISTERS
8 x 8
1
R1
0
R0
directly
addressable
when bank 1
is selected
addressed
indirectly
through pointers
R1 or R0
(R1' or R0')
directly
addressable
when bank 0
is selected
7Z84729

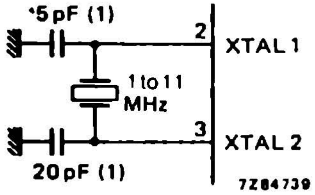
Fig. 4 Data memory map. In addition R0 or R1 (R0' or R1') may be used to address 256 words of an external RAM.

Program counter and stack

The program counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. When EA is '0', the PC can address locations 0 to 1023 of internal program memory. At the 1 K boundary, an automatic switch-over to external memory is made. When EA is '1', all the program is fetched from external ROM/EPROM. The total address space is 4 K bytes.
An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the program status word (PSW). Data RAM locations 8 to 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000B, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.
The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Oscillator and clock

The MAB8050H contains its own internal oscillator and clock driver. A crystal, inductor or external pulse generator determines the oscillator frequency (see Figs 5, 6 and 7). The output of the oscillator is divided-by-three and is available at T0 (pin 1) by executing the ENT0 CLK instruction. This CLK signal is divided-by-five to define a machine (instruction) cycle. It is available at ALE (pin 11).



(1) Including crystal-socket stray capacitance.

Fig. 5 Crystal oscillator mode. Crystal series impedance should be < 75 Ω at 6 MHz and < 180 Ω at 3,6 MHz.

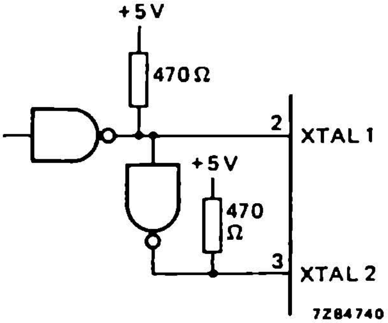
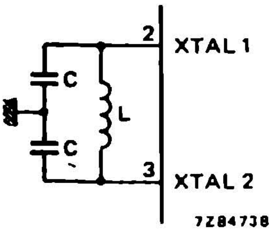


Fig. 6 Driving from external source. Both XTAL1 and XTAL2 should be driven. Resistors to V_{CC} (+ 5 V) are needed to ensure V_{IH} = 3,8 V if TTL circuitry is used. The minimum HIGH and LOW times are 45%.



L (μH)	C (pF)	nom. f (MHz)
45	20	5,2
120	20	3,2

$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3C_{pp}}{2}$$

Fig. 7 LC oscillator mode. Each C should be ≈ 20 pF including stray capacitance.
C_{pp} ≈ 5 to 10 pF (pin-to-pin capacitance).

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided-by-32 before they are applied to the input of the 8-bit counter. External events are applied directly to the input of the counter. The maximum frequency that can be counted is one third of the machine cycle frequency. The minimum positive duty cycle that can be detected is 0,2 times the cycle period. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

Interrupt

An interrupt may be generated by either an external input ($\overline{\text{INT}}$, pin 6) or the overflow of the internal counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine. After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

Input/output

The MAB8050H has 27 I/O lines. These lines are arranged as three 8-line ports, which serve as either inputs, outputs or bidirectional ports and 3 'test' inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, e.g., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The circuit configuration is shown in Fig. 8. Each line is continuously pulled up to + 5 V through a relative high resistance (≈ 50 kΩ). This pull-up is sufficient to provide the source current for a TTL HIGH level, yet can be pulled LOW by a standard TTL gate, thus allowing the same pin to be used for both input and output. To provide fast switching times during '0' to '1' transition a relatively low impedance transistor (≈ 5 kΩ) is switched on momentarily for 1/5 of a machine cycle whenever a '1' is written to the line. When a '0' is written, a low impedance (≈ 300 Ω) transistor overcomes the light pull-up and provides TTL current sinking capability.

Since the pull-down transistor is a low-impedance device, a '1' must first be written to any line which is to be used as an input. RESET initializes all lines to the high impedance '1' state. This structure allows input and output on the same pin and also allows a mixture of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

BUS

BUS is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.

As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{\text{RD}}$ and $\overline{\text{WR}}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{\text{WR}}$ output line and output data is valid at the trailing-edge of $\overline{\text{WR}}$. A read of the port generates a pulse on the $\overline{\text{RD}}$ output line and input data must be valid at the trailing-edge of $\overline{\text{RD}}$. When not being written or read, the BUS lines are high impedance.

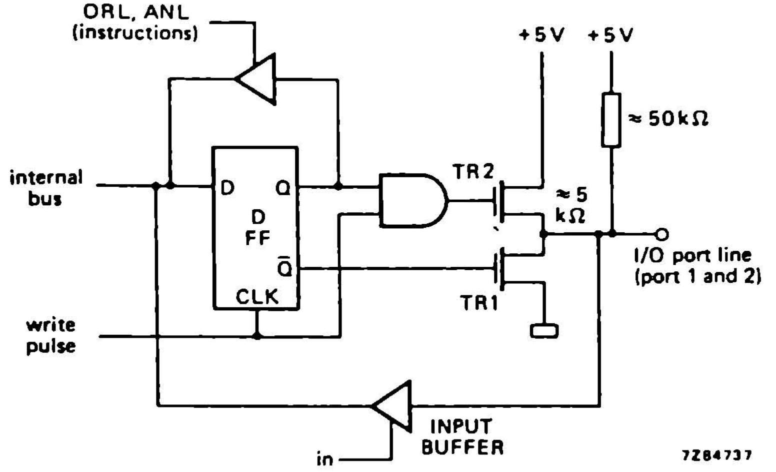


Fig. 8 Quasi-bidirectional port structure.

Test (T0, T1) and $\overline{INT}$ inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These pins are T0, T1, and $\overline{\text{INT}}$ and they allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1 and $\overline{\text{INT}}$ pins have other possible functions as well.

$\overline{\text{RESET}}$ input

The $\overline{\text{RESET}}$ input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pull-up resistor which, in combination with an external 1 μF capacitor, provides an internal reset pulse of sufficient duration to guarantee all circuitry is reset. If the reset pulse is generated externally, the $\overline{\text{RESET}}$ pin must be held at ground (0,45 V) for at least 10 ms after the power supply is within tolerance. Only five machine cycles (12,5 μs at 6 MHz) are required if power is already on and the oscillator has stabilized. Typical circuitry is shown in Fig. 9.

Single step input ($\overline{\text{SS}}$)

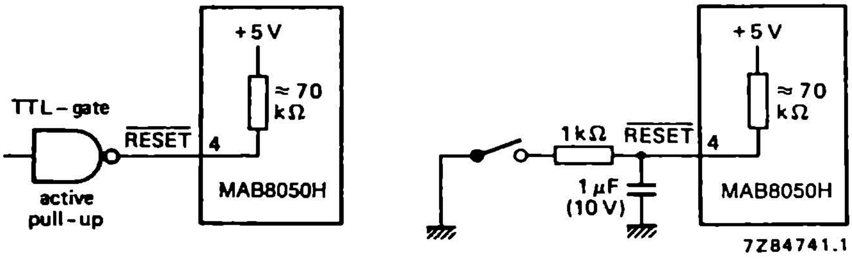
By proper control of the $\overline{\text{SS}}$ line, the processor can be forced to execute one instruction and then to wait until the single step switch is activated again.

Power-down mode

In the MAB8050H and MAB8040H , power can be removed from all but the 256x8 data RAM array, for low power standby operation. In the power-down mode the contents of the data RAM can be maintained while drawing typically 10% to 15% of the normal operating power supply voltage.

V_{CC} serves as the + 5 V supply pin for the bulk of the circuitry, while the V_{DD} pin supplies only the RAM array. In normal operation, both pins are at + 5 V. In standby, V_{CC} is at ground and only V_{DD} is maintained at + 5 V.

Applying $\overline{\text{RESET}}$ to the processor through the $\overline{\text{RESET}}$ pin inhibits any access to the RAM by the processor and guarantees that the RAM cannot be inadvertently altered as power is removed from V_{CC}. A typical power down sequence occurs as shown in Fig. 10.


+5 V
≈ 70 kΩ
TTL-gate
RESET
4
MAB8050H
active pull-up
+5 V
≈ 70 kΩ
1 kΩ
RESET
4
1 µF (10 V)
MAB8050H
7Z84741.1

(a) (b)

Fig. 9 An external reset circuit is shown in (a) and power-on reset in (b).

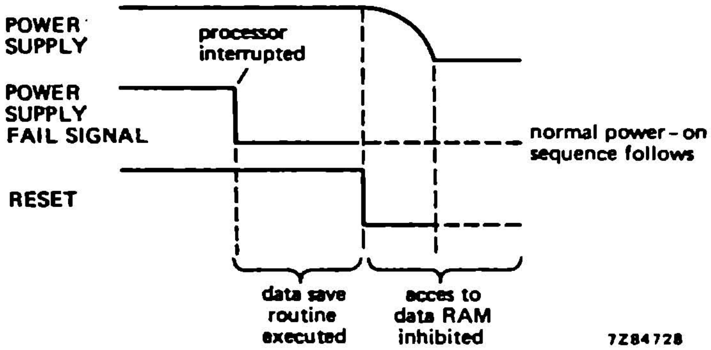

POWER SUPPLY
processor interrupted
POWER SUPPLY FAIL SIGNAL
normal power-on sequence follows
RESET
data save routine executed
acces to data RAM inhibited
7Z84728

Fig. 10 Power down sequence.

Instruction set

The MAB8050H instruction set consists of over 90 one and two-byte instructions (see Table 1). Program code efficiency is high because:

- working registers and program variables are stored in the RAM locations 0 to 255, which require only a single byte to address,
- program memory is divided into pages of 256 bytes, which means that branch destination addresses require one byte.

The instruction set efficiently manipulates and tests bits in addition to performing logical and arithmetic operations upon and the testing of bytes. A set of MOVE instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi-way branch (up to 256) upon the content of the accumulator to addresses stored in a look-up table. The 'decrement register and jump if not zero' instruction saves a byte every time it is used as opposed to using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The MAB8050H can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are highly desirable for real-time applications. See Table 2 for instruction timing.

Table 1 Instruction set is shown on the next 7 pages.

Symbols definitions used in Table 1.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number or expression (8-bits)
F0, F1	flags 0 and 1
I	interrupt
$\overline{\text{INT}}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1,	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
$	current value of program counter
←	is replaced by
↔	is exchanged with

Notes to Table 1.

1. Instruction code designations r and p form the binary representation of the registers and ports involved.
2. The dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

	mnemonic	function	description	instruction code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	cycles	bytes	flags C	AC	F0	F1	BS
ACCUMULATOR	ADD A,Rr	(A)←(A) + (Rr) for r = 0–7	Add contents of designated register to A	0	1	1	0	1	r	r	r	1	1	●	●			
	ADD A,@Rr	(A)←(A) + ((Rr)) for r = 0–1	Add indirect the contents of the data memory location to A	0	1	1	0	0	0	0	r	1	1	●	●			
	ADD A,#data	(A)←(A) + data	Add immediate data to A	0 d_7	0 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	●	●			
	ADDC A,Rr	(A)←(A) + (C) + (Rr) for r = 0–7	Add with carry the contents of designated register to A	0	1	1	1	1	r	r	r	1	1	●	●			
	ADDC A,@Rr	(A)←(A) + (C) + ((Rr)) for r = 0–1	Add indirect with carry the contents of the data memory location to A	0	1	1	1	0	0	0	r	1	1	●	●			
	ADDC A,#data	(A)←(A) + (C) + data	Add immediate data with carry to A	0 d_7	0 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	●	●			
	ANL A,Rr	(A)←(A) AND (Rr) for r = 0–7	Logical AND contents of designated register with A	0	1	0	1	1	r	r	r	1	1					
	ANL A,@Rr	(A)←(A) AND ((Rr)) for r = 0–1	Logical AND indirect the contents of data memory with A	0	1	0	1	0	0	0	r	1	1					
	ANL A,#data	(A)←(A) AND data	Logical AND immediate data with A	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
	CLR A	(A)←0	Clear the contents of A	0	0	1	0	0	1	1	1	1	1					
	CPL A	(A)←NOT(A)	Complement the contents of A	0	0	1	1	0	1	1	1	1	1					
	DA A		Decimal adjust the contents of A	0	1	0	1	0	1	1	1	1	1	●	●			
	DEC A	(A)←(A) −1	Decrement the contents of A by 1	0	0	0	0	0	1	1	1	1	1					
	INC A	(A)←(A) +1	Increment the contents of A by 1	0	0	0	1	0	1	1	1	1	1					
	ORL A,Rr	(A)←(A) OR (Rr) for r = 0–7	Logical OR contents of designated register with A	0	1	0	0	1	r	r	r	1	1					
	ORL A,@Rr	(A)←(A) OR ((Rr)) for r = 0–1	Logical OR indirect the contents of data memory location with A	0	1	0	0	0	0	0	r	1	1					

ACCUMULATOR (continued)	ORL A,#data	(A)←(A) OR data	Logical OR Immediate data with A	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2						
	RL A	(A_{n+1})←(A_n) (A_0)←(A_7) for n = 0–6	Rotate A left by 1-bit without carry	1	1	1	0	0	1	1	1	1	1						
	RLC A	(A_{n+1})←(A_n) (A_0)←(C) (C)←(A_7) for n = 0–6	Rotate A left by 1-bit through carry	1	1	1	1	0	1	1	1	1	1	●					
	RR A	(A_n)←(A_{n+1}) (A_7)←(A_0) for n = 0–6	Rotate A right by 1-bit without carry	0	1	1	1	0	1	1	1	1	1						
	RRC A	(A_n)←(A_{n+1}) (A_7)←(C) (C)←(A_0) n = 0–6	Rotate A right by 1-bit through carry	0	1	1	0	0	1	1	1	1	1	●					
	SWAP A	(A_{4-7})↔(A_{0-3})	Swap the two 4-bit nibbles in A	0	1	0	0	0	1	1	1	1	1						
	XRL A,Rr	(A)←(A) XOR (Rr)	Logical XOR contents of designated register with A	1	1	0	1	1	r	r	r	1	1						
	XRL A,@Rr	(A)←(A) XOR ((Rr))	Logical XOR indirect the contents of data memory location with A	1	1	0	1	0	0	0	r	1	1						
	XRL A,#data	(A)←(A) XOR data	Logical XOR Immediate data with A	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2						

	mnemonic	function	description	D7	D6	D5	D4	D3	D2	D1	D0	cycles	bytes	C	AC	F0	F1	BS
BRANCH	DJNZ Rr, addr	$(Rr) \leftarrow (Rr) - 1$ if (Rr) not zero: $(PC_{0-7}) \leftarrow$ addr	Decrement the specified register and test contents	1 a_7	1 a_6	1 a_5	0 a_4	1 a_3	r a_2	r a_1	r a_0	2	2					
	JBb addr	$(PC_{0-7}) \leftarrow$ addr if Bb = 1; $(PC) \leftarrow (PC) + 2$ if Bb = 0	Jump to specified address if accumulator bit is set	b_2 a_7	b_1 a_6	b_0 a_5	1 a_4	0 a_3	0 a_2	1 a_1	0 a_0	2	2					
	JC addr	$(PC_{0-7}) \leftarrow$ addr if C = 1; $(PC) \leftarrow (PC) + 2$ if C = 0	Jump to specified address if carry flag is set	1 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JF0 addr	$(PC_{0-7}) \leftarrow$ addr if (F0 = 1; $(PC) \leftarrow (PC) + 2$ if F0 = 0	Jump to specified address if flag F0 is set	1 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JF1 addr	$(PC_{0-7}) \leftarrow$ addr if F1 = 1; $(PC) \leftarrow (PC) + 2$ if F1 = 0	Jump to specified address if flag F1 is set	0 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JMP addr	$(PC_{8-10}) \leftarrow addr_{8-10}$ $(PC_{0-7}) \leftarrow addr_{0-7}$ $(PC_{11}) \leftarrow (DBF)$	Direct jump to specified address within the 2K address block	a_{10} a_7	a_9 a_6	a_8 a_5	0 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2					
	JMPP @A	$(PC_{0-7}) \leftarrow ((A))$	Jump indirect to specified address within address page	1	0	1	1	0	0	1	1	2	1					
	JNC addr	$(PC_{0-7}) \leftarrow$ addr if C = 0; $(PC) \leftarrow (PC) + 2$ if C = 1	Jump to specified address if carry flag is LOW	1 a_7	1 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNI	$(PC_{0-7}) \leftarrow$ addr if $\overline{INT}$ = 0; $(PC) \leftarrow (PC)$ + 2 if $\overline{INT}$ = 1	Jump to specified address if $\overline{INT}$ input is LOW	1 a_7	0 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNT0 addr	$(PC_{0-7}) \leftarrow$ addr if T0 = 0; $(PC) \leftarrow (PC) + 2$ if T0 = 1	Jump to specified address if T0 is LOW	0 a_7	0 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JNT1 addr	$(PC_{0-7}) \leftarrow$ addr if T1 = 0; $(PC) \leftarrow (PC) + 2$ if T1 = 1	Jump to specified address if T1 is LOW	0 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					

BRANCH (continued)	JNZ addr	(PC_{0-7})←addr if A ≠ 0; (PC)←(PC) + 2 if A = 0	Jump to specified address if A is non-zero	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JTF addr	(PC_{0-7})←addr if TF = 1; (PC)←(PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT0 addr	(PC_{0-7})←addr if T0 = 1; (PC)←(PC) + 2 if T0 = 0	Jump to specified address if T0 = 1	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT1 addr	(PC_{0-7})←addr if T1 = 1; (PC)←(PC) + 2 if T1 = 0	Jump to specified address if T1 = 1	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JZ addr	(PC_{0-7})←addr if A = 0; (PC)←(PC) + 2 if A ≠ 0	Jump to specified address if A is zero	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
CONTROL	EN I		Enable external ($\overline{INT}$) interrupt	0	0	0	0	0	1	0	1	1	1					
	DIS I		Disable external ($\overline{INT}$) interrupt	0	0	0	1	0	1	0	1	1	1					
	SEL RB0	(BS)←0	Select bank 0 (locations 0–7) of data memory	1	1	0	0	0	1	0	1	1	1					●
	SEL RB1	(BS)←1	Select bank 1 (locations 24–31) of data memory	1	1	0	1	0	1	0	1	1	1					●
	SEL MB0	(DBF)←0	Select program memory bank 0; addresses 0–2047	1	1	1	0	0	1	0	1	1	1					
	SEL MB1	(DBF)←1	Select program memory bank 1; addresses 2048–4095	1	1	1	1	0	1	0	1	1	1					
	ENT0 CLK		Enable clock output onto T0	0	1	1	1	0	1	0	1	1	1					

	mnemonic	function	description	instruction code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	cycles	bytes	flags C	AC	F0	F1	BS
DATA MOVES	MOV A,#data	(A)←data	Move immediate data into A	0 d_7	0 d_6	1 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
	MOV A,Rr	(A)←(Rr) for r = 0–7	Move the contents of the designated register into A	1	1	1	1	1	r	r	r	1	1					
	MOV A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of data memory into A	1	1	1	1	0	0	0	r	1	1					
	MOV A,PSW	(A)←(PSW)	Move contents of the program status word into A	1	1	0	0	0	1	1	1	1	1					
	MOV Rr,#data	(Rr)←data for r = 0–7	Move immediate data into the designated register	1 d_7	0 d_6	1 d_5	1 d_4	1 d_3	r d_2	r d_1	r d_0	2	2					
	MOV Rr,A	(Rr)←(A) for r = 0–7	Move A contents into the designated register	1	0	1	0	1	r	r	r	1	1					
	MOV @Rr,A	((Rr))←(A) for r = 0–1	Move indirect A contents into data memory location	1	0	1	0	0	0	0	r	1	1					
	MOV @Rr,#data	((Rr))←data for r = 0–1	Move indirect the specified data into data memory	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	r d_0	2	2					
	MOV PSW,A	(PSW)←A	Move contents of A into the program status word	1	1	0	1	0	1	1	1	1	1	●	●	●		●
	MOVP A,@A	(A)←((A))	Move data in the current page into A	1	0	1	0	0	0	1	1	2	1					
	MOVP3 A,@A	(A)←((A)) in page 3	Move data in page 3 into A	1	1	1	0	0	0	1	1	2	1					
	MOVX A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of external memory location into A	1	0	0	0	0	0	0	r	2	1					
	MOVX @Rr,A	((Rr))←(A) for r = 0–1	Move indirect the contents of A into external memory	1	0	0	1	0	0	0	r	2	1					
	XCH A,Rr	(A)↔(Rr) for r = 0–7	Exchange A with designated register contents	0	0	1	0	1	r	r	r	1	1					
	XCH A,@Rr	(A)↔((R)) for r = 0–1	Exchange indirect A contents with location in data memory	0	0	1	0	0	0	0	r	1	1					

D.M.	XCHD A,@Rr	$(A_{0-3}) \leftrightarrow (Rr_{0-3})$ for r = 0–1	Exchange indirect 4-bit contents of A with data memory	0	0	1	1	0	0	0	r	1	1					
FLAGS	CPL C	(C)←NOT (C)	Complement content of carry bit	1	0	1	0	0	1	1	1	1	1	●				
	CPL F0	(F0)←NOT (F0)	Complement content of flag F0	1	0	0	1	0	1	0	1	1	1			●		
	CPL F1	(F1)←NOT (F1)	Complement content of flag F1	1	0	1	1	0	1	0	1	1	1				●	
	CLR C	(C)←0	Clear content of carry bit to 0	1	0	0	1	0	1	1	1	1	1	●				
	CLR F0	(F0)←0	Clear content of flag F0 to 0	1	0	0	0	0	1	0	1	1	1			●		
	CLR F1	(F1)←0	Clear content of flag F1 to 0	1	0	1	0	0	1	0	1	1	1				●	
INPUT/OUTPUT	ANL BUS,#data	(BUS)←(BUS) AND data	Logical AND immediate data with BUS	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
	ANL Pp,#data	(Pp)←(Pp) AND data; p = 1–2	Logical AND immediate data with designated port (1 or 2)	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
	ANLD Pp,A	(Pp)←(Pp) AND (A_{0-3}); p = 4–7	Logical AND contents of A with designated port (4–7)	1	0	0	1	1	1	p	p	2	1					
	IN A,Pp	(A)←(Pp) p = 1–2	Input data from designated port (1–2) into A	0	0	0	0	1	0	p	p	2	1					
	INS A,BUS	(A)←(BUS)	Input strobed BUS data into A	0	0	0	0	1	0	0	0	1	2					
	MOVD A,Pp	(A_{0-3})←(Pp); p = 4–7 (A_{4-7})←0	Move contents of designated port (4–7) into A	0	0	0	0	1	1	p	p	2	1					
	MOVD Pp,A	(Pp)←(A_{0-3}) p = 4–7	Move contents of A to designated port (4–7)	0	0	1	1	1	1	p	p	1	1					
	ORLD Pp,A	(Pp)←(Pp) OR (A_{0-3}); p = 4–7	Logical OR contents of A with designated port (4–7)	1	0	0	0	1	1	p	p	1	1					
	ORL BUS,#data	(BUS)←(BUS) OR data	Logical OR immediate data with BUS	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
	ORL Pp,#data	(Pp)←(Pp) OR data p = 1–2	Logical OR immediate data with designated port (1–2)	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
	OUTL BUS,A	(BUS)←(A)	Output contents A onto BUS	0	0	0	0	0	0	1	0	1	2					
	OUTL Pp,A	(Pp)←(A) p = 1–2	Output contents A to designated port (1–2)	0	0	1	1	1	0	p	p	1	1					

	mnemonic	function	description	instruction code								cycles	bytes	flags				
				D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			C	AC	F0	F1	BS
REGISTER	DEC Rr	(Rr)←(Rr) −1 for r = 0–7	Decrement contents of designated register by 1	1	1	0	0	1	r	r	r	1	1					
	INC Rr	(Rr)←(Rr) + 1 for r = 0–7	Increment contents of designated register by 1	0	0	0	1	1	r	r	r	1	1					
	INC @Rr	((R))←((Rr)) + 1 for r = 0–1	Increment indirect the contents of data memory location by 1	0	0	0	1	0	0	0	r	1	1					
SUBROUTINE	CALL addr	((SP))←(PC), (PSW_{4-7}) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$	Call designated subroutine	a_{10} a_7	a_9 a_6	a_8 a_5	1 a_4	0 a_3	1 a_2	0 a_1	0 a_0	2	2					
	RET	(SP)←(SP) −1 (PC)←((SP))	Return from subroutine without restoring program status word	1	0	0	0	0	0	1	1	2	1					
	RETR	(SP)←(SP) −1 (PC)←((SP)) (PSW_{4-7})←((SP))	Return from subroutine restoring program status word	1	0	0	1	0	0	1	1	2	1					
TIMER/COUNTER	EN TCNTI		Enable timer/counter interrupt	0	0	1	0	0	1	0	1	1	1					
	DIS TCNTI		Disable timer/counter interrupt	0	0	1	1	0	1	0	1	1	1					
	MOV A,T	(A)←(T)	Move contents of timer/counter into A	0	1	0	0	0	0	1	0	1	1					
	MOV T,A	(T)←(A)	Move contents of A into timer/counter	0	1	1	0	0	0	1	0	1	1					
	STOP TCNT		Stop count for event counter or timer	0	1	1	0	0	1	0	1	1	1					
	STRT CNT		Start count for event counter	0	1	0	0	0	1	0	1	1	1					
	STRT T		Start count for timer	0	1	0	1	0	1	0	1	1	1					
	NOP		No operation	0	0	0	0	0	0	0	0	1	1					

Table 2 Instruction timing (see also Figs 11 and 12)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	fetch instruction	increment program counter	–	increment timer	–	–	read port	• –	–	–
OUTL P,A	¦	¦	–	¦	output to port	–	–	• –	–	–
ANL P,#data	¦	• ¦	–	¦	read port	fetch immediate data	–	•increment program counter	output to port	–
ORL P,#data	¦	• ¦	–	¦	read port	fetch immediate data	–	•increment program counter	output to port	–
INS A,BUS	¦	¦	–	¦	–	–	read port	• –	–	–
OUTL BUS,A	¦	¦	–	¦	output to port	–	–	• –	–	–
ANL BUS,#data	¦	• ¦	–	¦	read port	fetch immediate data	–	•increment program counter	output to port	–
ORL BUS,#data	¦	• ¦	–	¦	read port	fetch immediate data	–	•increment program counter	output to port	–
MOVX @R,A	¦	¦	output RAM address	¦	output data to RAM	–	–	• –	–	–
MOVX A,@R	¦	¦	output RAM address	¦	–	–	read data	• –	–	–
MOVD A,P	fetch instruction	increment program counter	output opcode/ address	increment timer	–	–	read P2 lower	• –	–	–

Continued on next page.

Table 2 Instruction timing (continued)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
MOVD P,A	fetch instruction	increment program counter	output opcode/ address	increment timer	output data to P2 lower	–	–	• –	–	–
ANLD P,A			output opcode/ address		output data	–	–	• –	–	–
ORLD P,A			output opcode/ address		output data	–	–	• –	–	–
J (conditional)		•	sample condition	increment timer	–	fetch immediate data	–	•update program counter	–	–
STRT CNT STRT T		•	–	–	start counter					
STOP TCNT		•	–	–	stop counter					
EN I		•	–	enable interrupt	–					
DIS I		•	–	disable interrupt	–					
ENT0 CLK	fetch instruction	•increment program counter	–	enable clock	–					

• Valid instruction addresses are output at this time if external program memory is being accessed.

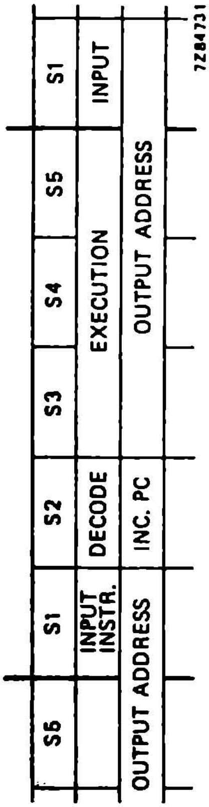

S5
S1
S2
S3
S4
S5
S1
INPUT INSTR.
DECODE
EXECUTION
INPUT
OUTPUT ADDRESS
INC. PC
OUTPUT ADDRESS
7Z84731

Fig. 11 Instruction cycle.

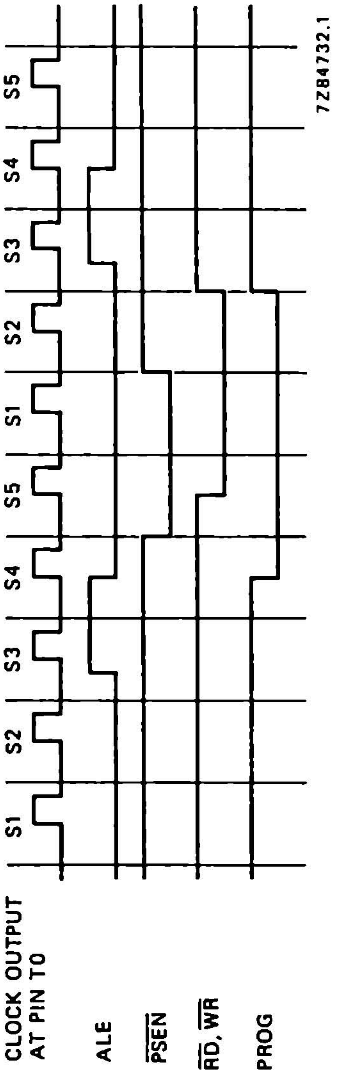

CLOCK OUTPUT AT PIN T0
ALE
PSEN
RD, WR
PROG
S1
S2
S3
S4
S5
S1
S2
S3
S4
S5
7Z84732.1

Fig. 12 Instruction cycle timing.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage with respect to V_{SS}			
except input EA	V_I	–0,5 to + 7	V
input EA	V_I	–0,5 to + 15	V
D.C. current into any input or output	$\pm I_I, \pm I_O$	max. 10	mA
Total power dissipation	P_{tot}	max. 1,5	W
Storage temperature range	T_{stg}	–65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit	conditions
Supply voltage	$V_{CC} = V_{DD}$	4,5	5,0	5,5	V	
Supply current	I_{DD}	–	–	9	mA	
Supply current (total)	$I_{CC} + I_{DD}$	–	–	90	mA	
Inputs						
Input voltage LOW all inputs except XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IL}	–0,5	–	0,8	V	
Input voltage LOW XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IL1}	–0,5	–	0,6	V	
Input voltage HIGH all inputs except XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IH}	2,0	–	V_{CC}	V	
Input voltage HIGH XTAL1, XTAL2, $\overline{\text{RESET}}$	V_{IH1}	3,8	–	V_{CC}	V	
Outputs						
Output voltage LOW; BUS	V_{OL}	–	–	0,45	V	I_{OL} = 2 mA
Output voltage LOW $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE	V_{OL1}	–	–	0,45	V	I_{OL1} = 2 mA
Output voltage LOW; PROG	V_{OL2}	–	–	0,45	V	I_{OL2} = 1 mA
Output voltage LOW all other outputs	V_{OL3}	–	–	0,45	V	I_{OL3} = 1,6 mA
Output voltage HIGH; BUS	V_{OH}	2,4	–	–	V	$-I_{OH}$ = 100 μA
Output voltage HIGH $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE	V_{OH1}	2,4	–	–	V	$-I_{OH1}$ = 100 μA
Output voltage HIGH all other outputs	V_{OH2}	2,4	–	–		$-I_{OH2}$ = 50 μA

parameter	symbol	min.	typ.	max.	unit	conditions
Input leakage current T1, $\overline{\text{INT}}$	± I_{IL}	–	–	10	μA	$V_{SS} < V_I < V_{CC}$
Input leakage current P10 to P17, P20 to P27, EA, $\overline{\text{SS}}$	$-I_{IL1}$	–	–	500	μA	$V_{SS} + 0{,}45\ V < V_I < V_{CC}$
Output leakage current BUS, T0 (high impedance)	± I_{OL}	–	–	10	μA	$V_{SS} + 0{,}45\ V < V_I < V_{CC}$

A.C. CHARACTERISTICS

$V_{CC} = V_{DD} = 5$ V ± 10%; $V_{SS} = 0$ V; $T_{amb} = 0$ to + 70 °C

See waveforms Figs 13, 14 and 15

parameter	symbol	f = 6 MHz min.	f = 6 MHz max.	f = 11 MHz min.	f = 11 MHz max.	unit	conditions note 1
ALE pulse width	t_{LL}	400	–	150	–	ns	
Address set-up time to ALE	t_{AL}	150	–	70	–	ns	
Address hold time from ALE	t_{LA}	80	–	50	–	ns	
Control pulse width $\overline{\text{PSEN}}$, $\overline{\text{RG}}$, $\overline{\text{WR}}$	t_{CC}	700	–	300	–	ns	
Data set-up time before $\overline{\text{WR}}$	t_{DW}	500	–	250	–	ns	
Data hold time after $\overline{\text{WR}}$	t_{WD}	120	–	40	–	ns	note 2
Cycle time	t_{CY}	2,5	15,0	1,36	15.0	μs	
Data hold time	t_{DR}	0	200	0	100	ns	
$\overline{\text{PSEN}}$, $\overline{\text{RD}}$ to data input	t_{RD}	–	500	–	200	ns	
Address set-up time to $\overline{\text{WR}}$	t_{AW}	230	–	200	–	ns	
Address set-up time to data input	t_{AD}	–	950	–	400	ns	
Address floating to $\overline{\text{RD}}$, $\overline{\text{PSEN}}$	t_{AFC}	0	–	–10	–	ns	

Notes

1. Control outputs: $C_L = 80$ pF
   Bus outputs: $C_L = 150$ pF
2. Bus high-impedance load: 20 pF

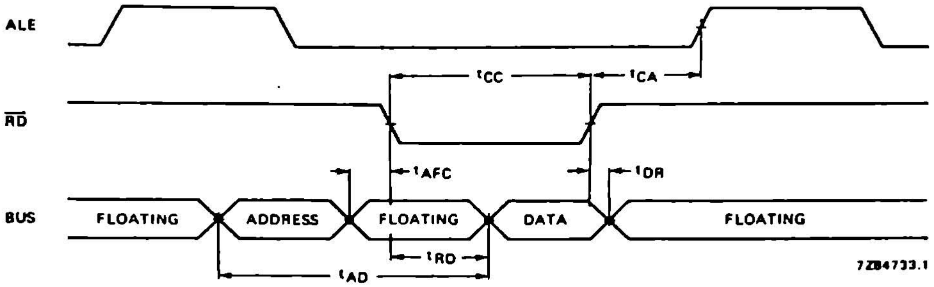

ALE
$\overline{\text{RD}}$
BUS
t_{CC}
t_{CA}
t_{AFC}
t_{DR}
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
t_{RD}
t_{AD}
7Z84733.1

Fig. 13 Read from external data memory.

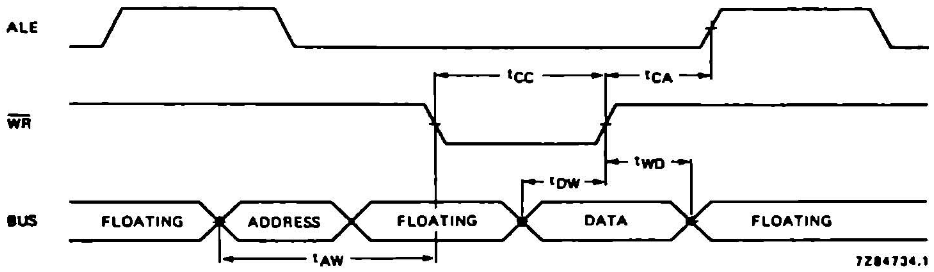

ALE
$\overline{\text{WR}}$
BUS
t_{CC}
t_{CA}
t_{WD}
t_{DW}
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
t_{AW}
7Z84734.1

Fig. 14 Write to external data memory.

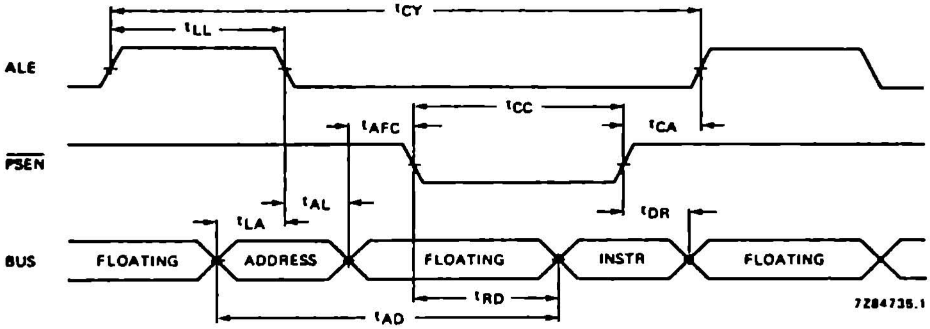

t_{CY}
t_{LL}
ALE
t_{CC}
t_{AFC}
t_{CA}
$\overline{\text{PSEN}}$
t_{AL}
t_{LA}
t_{DR}
BUS
FLOATING
ADDRESS
FLOATING
INSTR
FLOATING
t_{RD}
t_{AD}
7Z84735.1

Fig. 15 Instruction fetch from external program memory.

A.C. CHARACTERISTICS

Port 2 timing (see Fig. 16)

$V_{CC} = 5\ V \pm 10\%$; $V_{SS} = 0\ V$; $T_{amb} = 0$ to $+ 70\ ^oC$

parameter	symbol	f = 6 MHz min.	f = 6 MHz max.	f = 11 MHz min.	f = 11 MHz max.	unit
Port control set-up time before falling edge of PROG	t_{CP}	110	–	60	–	ns
Port control hold time after falling edge of PROG	t_{PC}	100	–	55	–	ns
PROG to time P_2 input must be valid	t_{PR}	–	810	–	440	ns
Input data hold time	t_{PF}	0	150	0	80	ns
Output data set-up time	t_{DP}	250	–	130	–	ns
Output data hold time	t_{PD}	65	–	35	–	ns
PROG pulse width	t_{PP}	1200	–	650	–	ns
Port 2 I/O data set-up time	t_{PL}	350	–	200	–	ns
Port 2 I/O data hold time	t_{LP}	150	–	80	–	ns

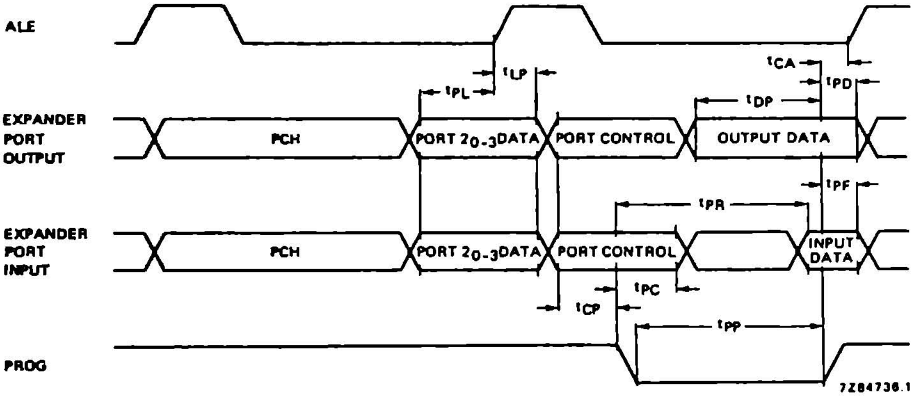
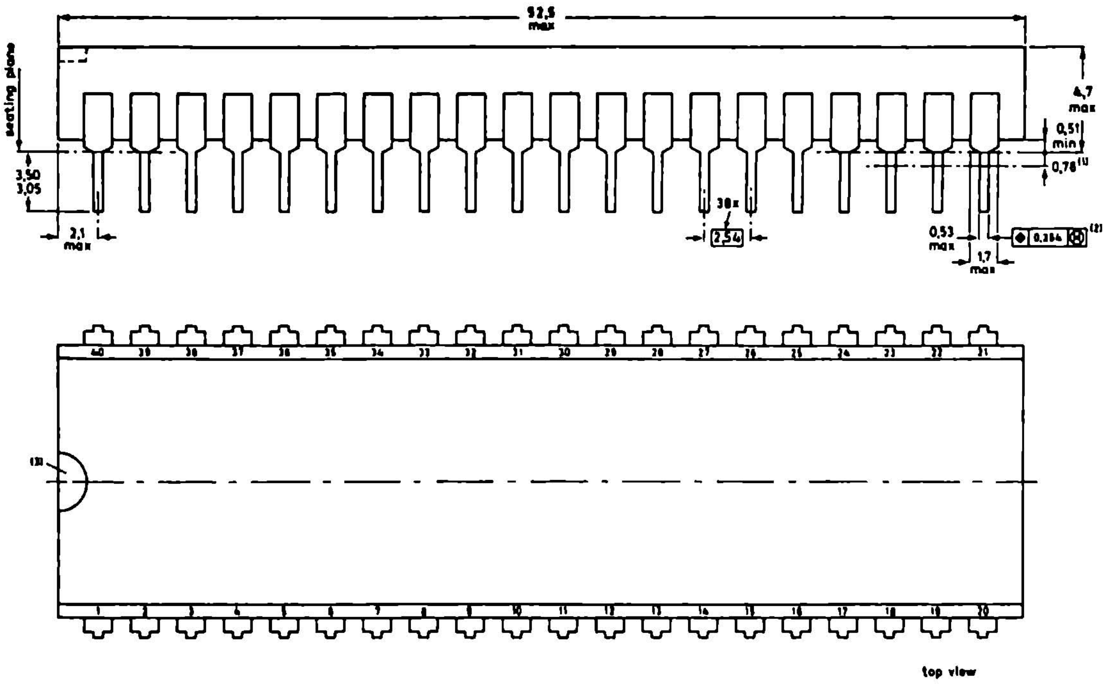
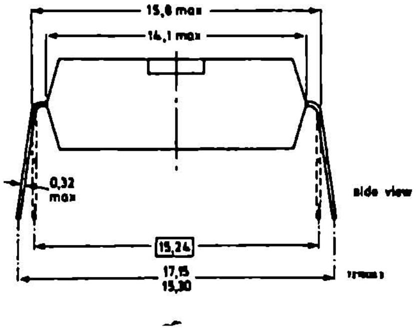

ALE
tLP
tPL
tCA
tPD
tDP
EXPANDER PORT OUTPUT
PCH
PORT 2₀-₃DATA
PORT CONTROL
OUTPUT DATA
tPF
tPR
EXPANDER PORT INPUT
PCH
PORT 2₀-₃DATA
PORT CONTROL
INPUT DATA
tPC
tCP
tPP
PROG
7Z84736.1

Fig. 16 Port 2 timing.

40-LEAD DUAL IN-LINE; PLASTIC (SOT-129)


92,5 max
seating plane
3,50
3,05
2,1 max
38x
2,54
4,7 max
0,51 min
0,78 (1)
0,53 max
1,7 max
0,254 Ⓜ (2)
40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
(3)
top view


15,8 max
14,1 max
0,32 max
15,24
17,15
15,90
side view

Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

SOLDERING

See next page.

SOLDERING

1. By hand

Apply the soldering iron below the seating plane (or not more than 2 mm above it).
If its temperature is below 300 °C it must not be in contact for more than 10 seconds; if between 300 °C and 400 °C, for not more than 5 seconds.

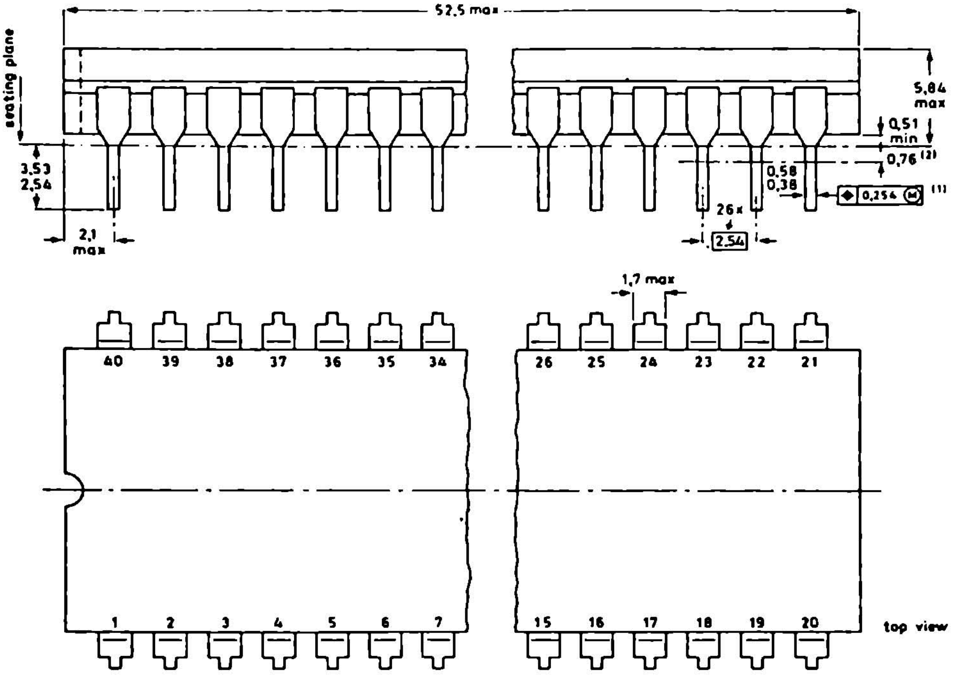
2. By dip or wave

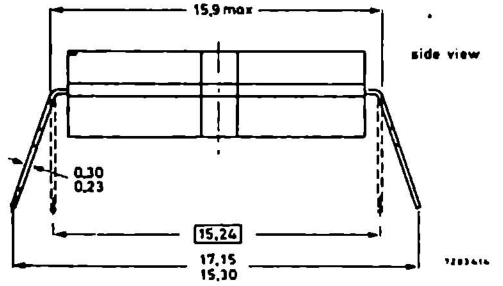
The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.
The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

3. Repairing soldered joints

The same precautions and limits apply as in (1) above.

40-LEAD DUAL IN-LINE; CERAMIC (SOT-145)


52,5 max
seating plane
5,84 max
0,51 min
0,76 (2)
3,53
2,54
0,58
0,38
0,254 Ⓜ (1)
26×
2,54
2,1 max
1,7 max
40 39 38 37 36 35 34
26 25 24 23 22 21
1 2 3 4 5 6 7
15 16 17 18 19 20
top view


15,9 max
side view
0,30
0,23
15,24
17,15
15,30

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

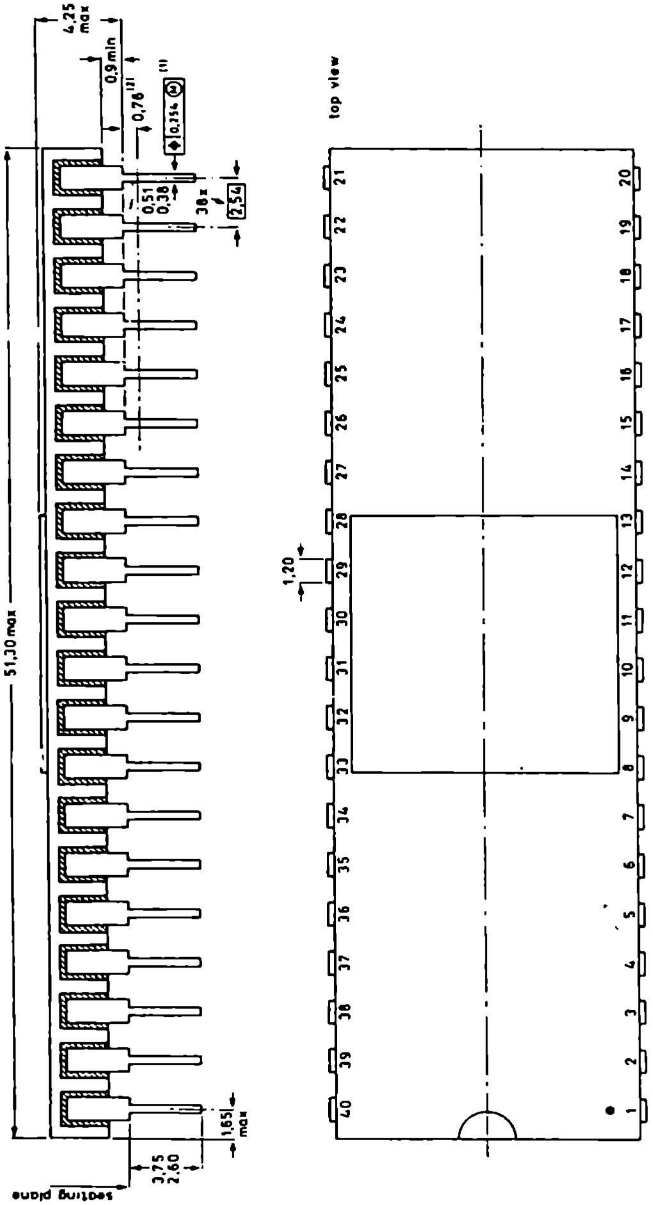
(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

Remarks

1. Leads are given positive misalignment so that they grip after insertion.
2. Leads are Ni-Fe, pure tin plated.

40-LEAD DUAL IN-LINE; METAL-CERAMIC (SOT-88B)


4,25 max
0,9 min
0,76 (2)
0,254 Ⓜ (1)
0,51
0,38
38x
2,54
51,30 max
1,65 max
3,75
2,60
seating plane
top view
1,20
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

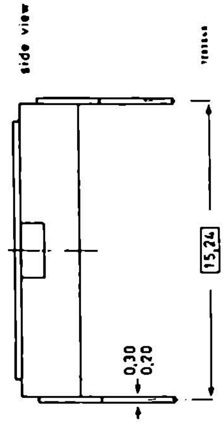
⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm


side view
0,30
0,20
15,24

SINGLE-CHIP 8-BIT MICROCOMPUTERS

DESCRIPTION

The MAB8400 family of single-chip 8-bit microcomputers are fabricated in N-MOS.

The family consists of 4 devices:

- **MAB8400**: capacity of 128 RAM bytes ('Piggy-back version'),
- **MAB8410**: 1K ROM/64 RAM bytes,
- **MAB8420**: 2K ROM/64 RAM bytes, x)
- **MAB8440**: 4K ROM/128 RAM bytes.

Each type has 20 quasi-bidirectional I/O port lines, one serial I/O line, one single-level vectored interrupt, and an 8-bit timer/event counter and an on-board clock oscillator and clock circuits.

This microcomputer family is designed to be an efficient controller as well as an arithmetic processor. The instruction set is based on that of the MAB8048. The microcomputers have extensive bit handling ability and facilities for both binary and BCD arithmetic.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead DIL package
- 1K, 2K or 4K ROM bytes
- 64, 64 or 128 RAM bytes
- 20 quasi-bidirectional I/O port lines
- Two test inputs: one of which can be used to detect zero voltage cross-over, the other is also the external interrupt input
- Single-level vectored interrupts: external, timer/event counter, serial I/O
- Serial I/O which can be used in bus systems with more than one master (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Internal oscillator, clock driver
- Over 80 instructions (based on MAB8048)
- All instructions 1 or 2 cycles
- Single 5 V supply (± 10%)
- Operating temperature range: 0 to + 70 °C (MAB8400 family), or –40 to + 85 °C (MAF8400 family)

x) MAB 8421: same as MAB 8420,
but Port 1: $V_{OL} \leqq 1{,}0$ V at I_{OL} = max. 10 mA

PACKAGE OUTLINES

MAB8400B: 28-lead 'Piggy-back' package (with up to 28-lead EPROM on top).
MAB8400Q: 56-lead QUIL; plastic (VO-35).
MAB8410P/20P/40P: 28-lead DIL; plastic (SOT-117D).
MAB8410D/20D/40D: 28-lead DIL; ceramic (SOT-135).
MAB8410T/20T/40T: 28-lead flat pack; plastic (SO-28; SOT-136A).

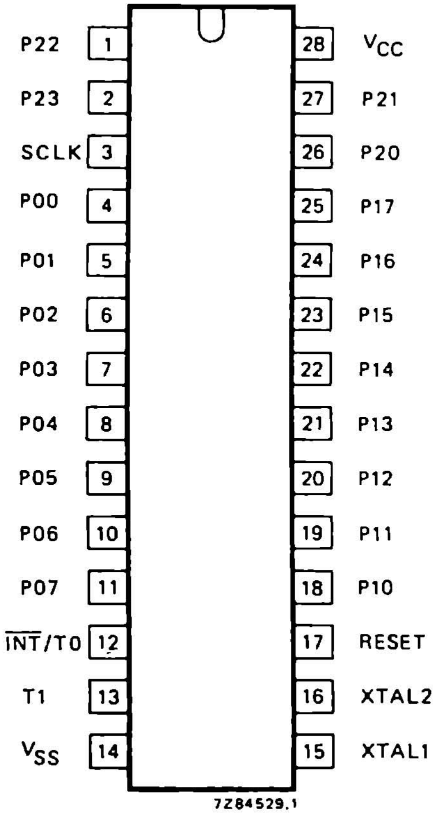

P22 1
P23 2
SCLK 3
P00 4
P01 5
P02 6
P03 7
P04 8
P05 9
P06 10
P07 11
$\overline{\text{INT}}$/T0 12
T1 13
V_{SS} 14
28 V_{CC}
27 P21
26 P20
25 P17
24 P16
23 P15
22 P14
21 P13
20 P12
19 P11
18 P10
17 RESET
16 XTAL2
15 XTAL1
7Z84529.1

Fig. 1 Pinning diagram for MAB8400B 'Piggy-back' version bottom pinning (for top pinning see Fig. 2), MAB8410, MAB8420 and MAB8440.

PIN DESIGNATION

designation	pin no.	function
V_{SS}	14	Ground
V_{CC}	28	Power supply, + 5 V.
P00-P07	4-11	Port 0. 8-bit quasi-bidirectional I/O port.
P10-P17	18-25	Port 1. 8-bit quasi-bidirectional I/O port.
P20-P23	26,27,1,2	Port 2. 4 bit quasi-bidirectional I/O port, P23 is the serial data input/output in serial I/O mode.
SCLK	3	Bidirectional clock for serial I/O.
$\overline{\text{INT}}$/T0	12	External interrupt input (sensitive to negative-going edge); testable using the JT0, JNT0 instructions.
T1	13	Input pin testable using the JT1, JNT1 instructions. Can be designated the event counter input, using the STRT CNT instruction. Also allows zero cross-over sensing of slowly moving a.c. inputs.
RESET	17	Input, used to initialize the processor (active HIGH).
XTAL1	15	Connection to timing component (crystal) which determines the frequency of the internal oscillator. Also the input for an external clock source.
XTAL2	16	Connection of the other side of timing component.

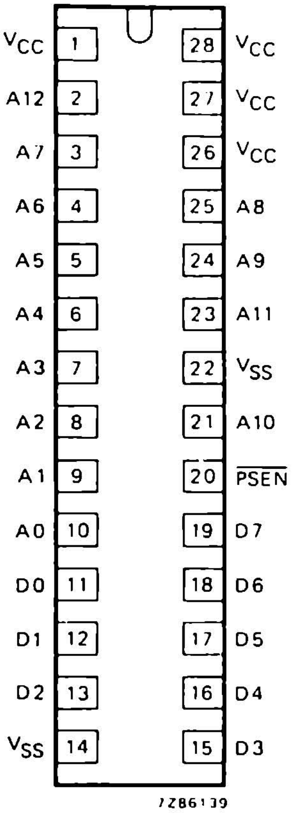

V_{CC} 1
A12 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
D0 11
D1 12
D2 13
V_{SS} 14
28 V_{CC}
27 V_{CC}
26 V_{CC}
25 A8
24 A9
23 A11
22 V_{SS}
21 A10
20 $\overline{PSEN}$
19 D7
18 D6
17 D5
16 D4
15 D3
7Z86139

Fig. 2 Pinning diagram for MAB8400B 'Piggy-back' version top pinning (for bottom pinning see Fig. 1); to access a 2732 or 2764 EPROM.

PIN DESIGNATION

designation	pin no.	function
V_{SS}	14, 22	Ground
V_{CC}	1, 26-28	Power supply, + 5 V
A0-A12	10-3, 25, 24, 21, 23, 2	Address outputs
D0-D7	11-13, 15-19	Data
$\overline{PSEN}$	20	Program store enable

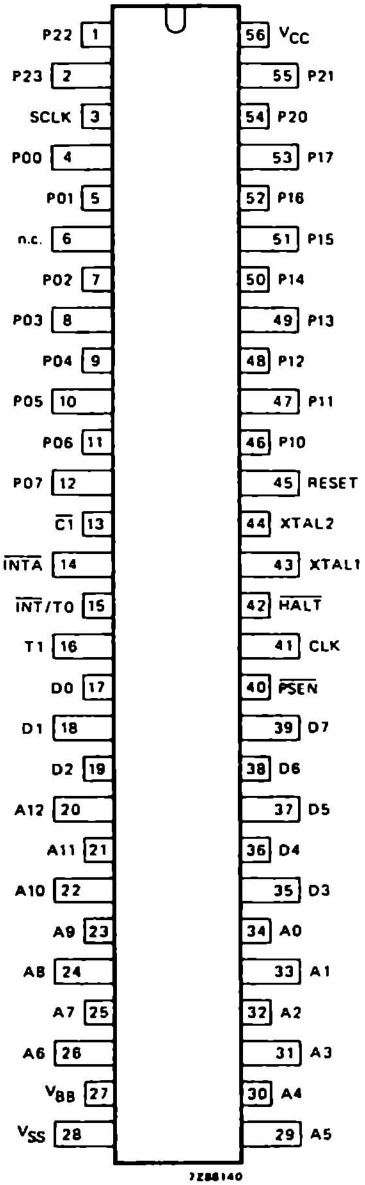

P22 1
P23 2
SCLK 3
P00 4
P01 5
n.c. 6
P02 7
P03 8
P04 9
P05 10
P06 11
P07 12
C̄1 13
INTA 14
INT/T0 15
T1 16
D0 17
D1 18
D2 19
A12 20
A11 21
A10 22
A9 23
A8 24
A7 25
A6 26
VBB 27
VSS 28
56 VCC
55 P21
54 P20
53 P17
52 P16
51 P15
50 P14
49 P13
48 P12
47 P11
46 P10
45 RESET
44 XTAL2
43 XTAL1
42 HALT
41 CLK
40 PSEN
39 D7
38 D6
37 D5
36 D4
35 D3
34 A0
33 A1
32 A2
31 A3
30 A4
29 A5
7Z88140

Fig. 3 Pinning diagram for MAB8400Q; 56-leads, quadruple in-line.

PIN DESIGNATION

designation	pin no.	function
V_{SS}	28	Ground
V_{CC}	56	Power supply, + 5 V
P00-P07	4, 5, 7-12	Port 0. 8-bit quasi-bidirectional I/O port.
P10-P17	46-53	Port 1. 8-bit quasi-bidirectional I/O port.
P20-P23	54, 55, 1, 2	Port 2. 4-bit quasi-bidirectional I/O port; P23 is the serial data input/output in serial I/O mode.
SCLK	3	Bidirectional clock for serial I/O.
$\overline{\text{INT}}/\overline{\text{T0}}$	15	External interrupt input (sensitive to negative-going edge); testable using the JT0, JNT0 instructions.
T1	16	Input pin testable using the JT1, JNT1 instructions. Can be designated the event counter input, using the STRT CNT instruction. This also allows zero cross-over sensing of slowly moving a.c. inputs.
RESET	45	Input, used to initialize the processor (active HIGH).
XTAL1	43	Connection to timing component (crystal) which determines the frequency of the internal oscillator. Also the input for an external clock source.
XTAL2	44	Connection of the other side of timing component.
V_{BB}	27	Back bias voltage.

PIN DESIGNATION (continued)

designation	pin no.	function
A0-A12	34-29, 26-20	Program memory address outputs (active HIGH); A0 = LSB, A12 = MSB. Address output change after begin Phi3 of TS8.
D0-D7	17-19, 35-39	Data input lines (active HIGH). Used for reading external program memory. D0 = LSB, D7 = MSB.
CLK	41	Clock output buffered from XTAL2. On the positive-going edge the (internal) Phi clock goes HIGH.
$\overline{\text{PSEN}}$	40	Program store enable. This signal is used for enabling the external EPROM (e.g. on the 'Piggy-back' version). For emulation it enables the emulation memory and it indicates machine cycles. Active LOW during TS9, TS10 of each machine cycle and TS1 of the following machine cycle.
$\overline{\text{C1}}$	13	Cycle 1 indication output (active LOW). During emulation this signal indicates the opcode fetch cycle (useful for external instruction decoding, real time trace). Active from begin of TS10 of the cycle preceeding cycle 1, to begin TS10 of cycle 1.
$\overline{\text{HALT}}$	42	Halt input (active LOW). If activated, the current instruction is finished and the microcomputer stops execution (HALT mode). The next following program counter address is available on the address bus. Program counter and timer/counter are no longer updated. The serial I/O finishes the current transmit/receive action and goes into the idle state. Interrupts are not sampled in the HALT mode, they are only sampled when the microcomputer is running. Interrupt routines can be single-stepped as a normal program.
$\overline{\text{INTA}}$	14	Interrupt acknowledge output (active LOW). It indicates the acception of any interrupt. Active from begin of TS8 of the interrupted cycle, to begin of TS7 of the second cycle of the (internally forced) 'CALL vector address' instruction. During $\overline{\text{INTA}}$ active, the address bus shows the address, that has been saved in the stack (return address); the $\overline{\text{C1}}$ output indicates opcode fetch cycles as if a user CALL was executed.

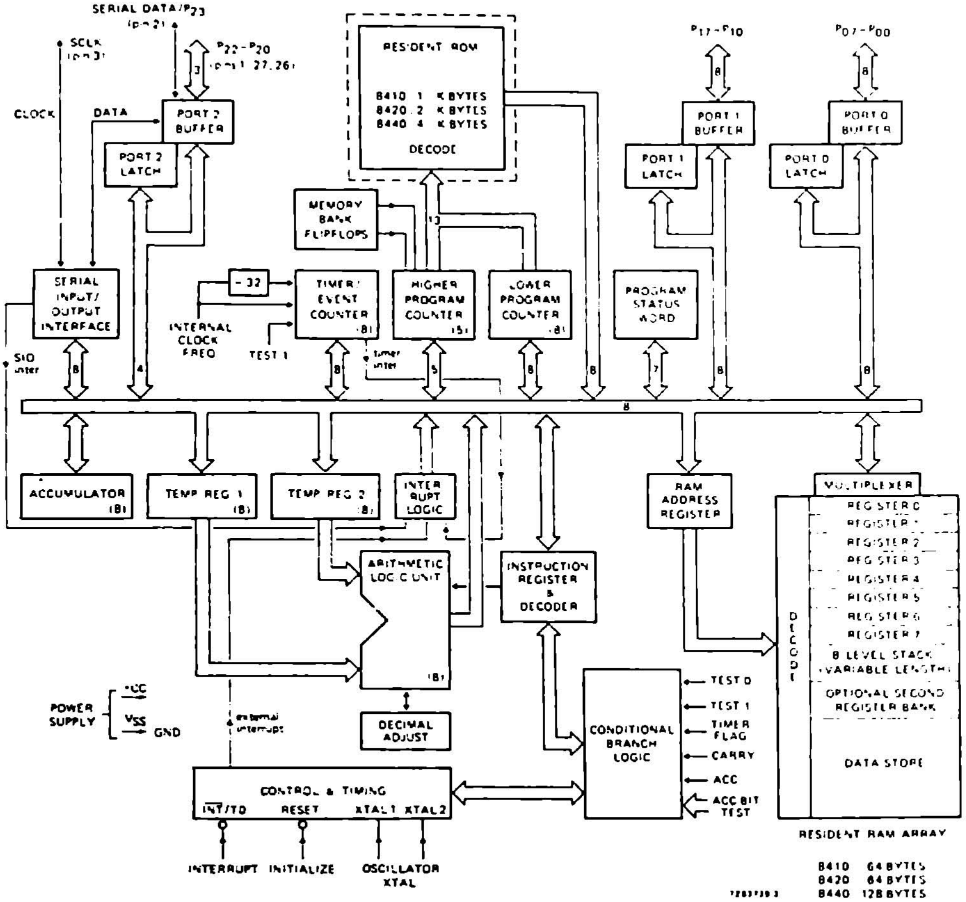

SERIAL DATA/P_{23}
(pin 2)
SCLK
(pin 3)
$P_{22}-P_{20}$
(pins 1, 27, 26)
3
CLOCK
DATA
PORT 2
BUFFER
PORT 2
LATCH
RESIDENT ROM
8410 1 K BYTES
8420 2 K BYTES
8440 4 K BYTES
DECODE
MEMORY
BANK
FLIPFLOPS
13
$P_{17}-P_{10}$
8
PORT 1
BUFFER
PORT 1
LATCH
$P_{07}-P_{00}$
8
PORT 0
BUFFER
PORT 0
LATCH
SERIAL
INPUT/
OUTPUT
INTERFACE
÷ 32
TIMER/
EVENT
COUNTER
(8)
INTERNAL
CLOCK
FREQ
TEST 1
HIGHER
PROGRAM
COUNTER
(5)
LOWER
PROGRAM
COUNTER
(8)
PROGRAM
STATUS
WORD
SIO
inter
8
4
8
timer
inter
5
8
8
7
8
8
8
ACCUMULATOR
(8)
TEMP REG 1
(8)
TEMP REG 2
(8)
INTER
RUPT
LOGIC
RAM
ADDRESS
REGISTER
MULTIPLEXER
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
8 LEVEL STACK
(VARIABLE LENGTH)
OPTIONAL SECOND
REGISTER BANK
DATA STORE
DECODE
ARITHMETIC
LOGIC UNIT
(8)
INSTRUCTION
REGISTER
&
DECODER
POWER
SUPPLY
V_{CC}
V_{SS}
GND
external
interrupt
DECIMAL
ADJUST
CONDITIONAL
BRANCH
LOGIC
TEST 0
TEST 1
TIMER
FLAG
CARRY
ACC
ACC BIT
TEST
CONTROL & TIMING
$\overline{INT}$/T0
RESET
XTAL 1
XTAL 2
INTERRUPT
INITIALIZE
OSCILLATOR
XTAL
RESIDENT RAM ARRAY
8410 64 BYTES
8420 64 BYTES
8440 128 BYTES
7Z83739.3

Fig. 4a Block diagram of the MAB8400 family.

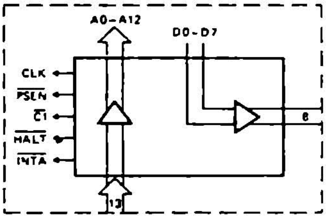

A0–A12
D0–D7
CLK
$\overline{PSEN}$
$\overline{C1}$
$\overline{HALT}$
$\overline{INTA}$
8
13

Fig. 4b Replacement of dotted part in Fig. 4a, showing the MAB8400Q bond-out version.

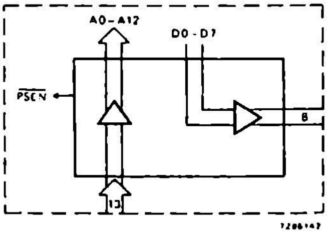

A0–A12
D0–D7
$\overline{PSEN}$
8
13
7Z80147

Fig. 4c Replacement of dotted part in Fig. 4a, for the MAB8400B 'Piggy-back' version.

FUNCTIONAL DESCRIPTION

Bond-out version MAB8400Q

The bond-out version is a microcomputer that contains no on-board ROM, but has all address and data lines brought-out to access an external ROM or EPROM. So, this version has more pins than the standard microcomputers with on-board ROM. It has all the features of the other members of the MAB8400 family. The RAM has 128 bytes.

Program memory (ROM)

The program memory consists of 1024, 2048, or 4096 bytes (8-bit words), which are addressed by the program counter. The memory is mask-programmed at our factory. Because the MAB8400 family offers a range of ROM capacities to suit the application, ROM expansion is not required. Figure 5 shows the program memory map.
Four program memory locations are of special importance:

location 0 – contains the first instruction to be executed after the processor is initialized (RESET),
location 3 – contains the first byte of an external interrupt service subroutine,
location 5 – contains the first byte of a serial I/O interrupt service subroutine,
location 7 – contains the first byte of a timer/event counter interrupt service subroutine.

Program memory is arranged in banks of 2K bytes, which are selected by SEL MB instructions. Further, the program memory is organized in pages of 256 bytes. Only the unconditional branch instructions (JMP and CALL) can cause jumps over page boundaries. Memory bank boundaries can be crossed only by using the same unconditional branch instructions after the appropriate memory bank has been selected. A CALL instruction can transfer control to a subroutine on any page; RET and RETR instructions can transfer control from a subroutine back to the main program. Note, a memory bank must be selected prior to a CALL or JMP instruction.

Data memory (RAM)

Data memory consists of 64 or 128 bytes (8-bit words). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 6 shows the data memory map.

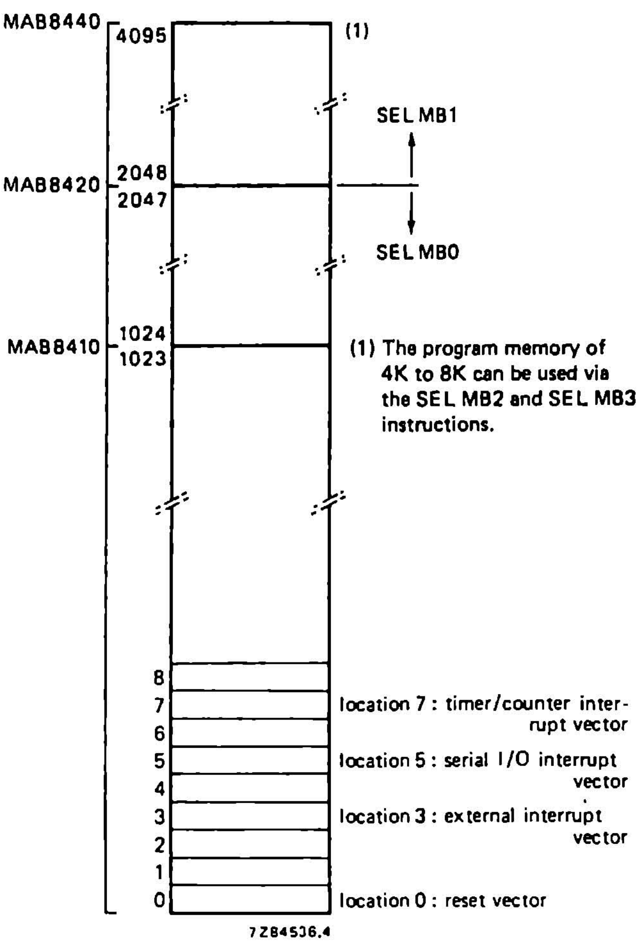
Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Because these registers are easily addressed and require the minimum instruction bytes to manipulate their contents, they are commonly used to store frequently accessed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines, saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first two locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 7) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the program counter stack's eight register pairs will be loaded with the next return address generated.


MAB8440
MAB8420
MAB8410
4095
2048
2047
1024
1023
8
7
6
5
4
3
2
1
0
(1)
SEL MB1
SEL MB0
location 7 : timer/counter interrupt vector
location 5 : serial I/O interrupt vector
location 3 : external interrupt vector
location 0 : reset vector
7Z84536.4

(1) The program memory of 4K to 8K can be used via the SEL MB2 and SEL MB3 instructions.

Fig. 5 Program memory map.

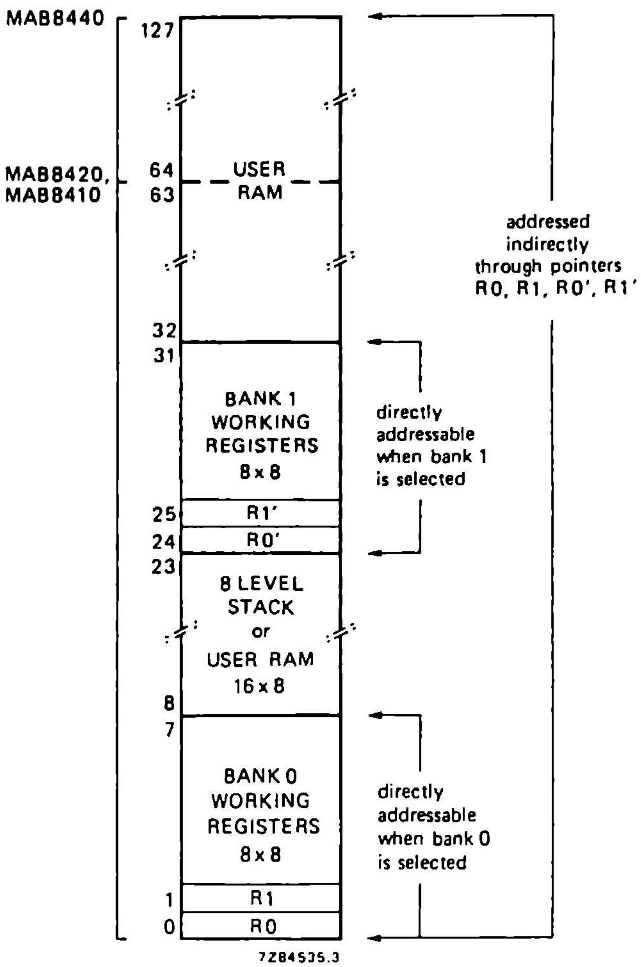

MAB8440
MAB8420,
MAB8410
127
64
63
32
31
25
24
23
8
7
1
0
USER
RAM
BANK 1
WORKING
REGISTERS
8 x 8
R1'
R0'
8 LEVEL
STACK
or
USER RAM
16 x 8
BANK 0
WORKING
REGISTERS
8 x 8
R1
R0
addressed
indirectly
through pointers
R0, R1, R0', R1'
directly
addressable
when bank 1
is selected
directly
addressable
when bank 0
is selected
7Z84535.3

Fig. 6 Data memory map.

FUNCTIONAL DESCRIPTION (continued)

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another call. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations. Possible locations from 32 to 127 may be used for storage of program variables or data.

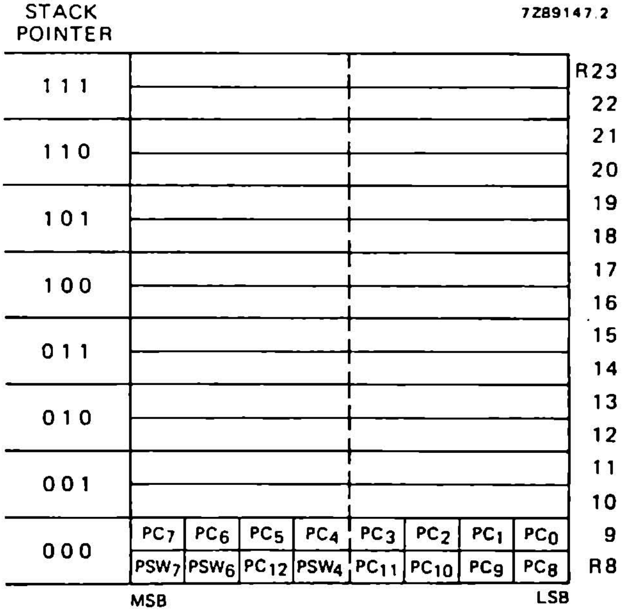

STACK POINTER
7Z89147.2
111
110
101
100
011
010
001
000
R23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
R8
PC_7 PC_6 PC_5 PC_4 PC_3 PC_2 PC_1 PC_0
PSW_7 PSW_6 PC_{12} PSW_4 PC_{11} PC_{10} PC_9 PC_8
MSB
LSB

Fig. 7 Program counter stack.

Input/output

The MAB8400 family has 23 I/O lines arranged as:
- two parallel ports of 8 lines (P00 to P07, P10 to P17),
- one parallel port of 4 lines (P20 to P23),
- a serial I/O consisting of a data line shared with a parallel port line (P23) and a separate clock line SCLK,
- one external interrupt and test input ($\overline{\text{INT}}$/T0); when used as a test input it can be tested by the conditional branch instructions JT0 and JNT0,
- one test input (T1), which can alter program sequences when tested by conditional jump instructions JT1 and JNT1; T1 can also be used as an input to the timer/event counter, or for zero-cross detection.

FUNCTIONAL DESCRIPTION (continued)

Parallel ports

Output data written to a port is latched and remains unchanged until rewritten. Input data is not latched and so must be present until read by an input instruction.

Input lines are fully TTL compatible, output lines can drive one standard TTL load. Figure 8a shows the quasi-bidirectional I/O interface with push-pull output with pull-up resistance. Each line is continuously pulled up to + 5 V through a relatively high resistance (≈ 50 kΩ). When a '0' is written to the line, the low impedance of TR1 overcomes the pull-up and provides TTL current sinking capability. When a '1' is written, TR2 is momentarily switched on to give fast pull-up. One state of a machine cycle later, the '1' level is maintained by the pull-up through the 50 kΩ. When used as an input line, a '1' must first be written to the line, otherwise the pull-down transistor TR1 is low impedance. This can be done by software control. After RESET, all I/O lines are in the input mode. The '1' on these lines can then be easily pulled down to a '0' by CMOS or TTL circuits.

Two other interface output configurations can also be specified:
- open drain output with pull-up (Fig. 8b),
- open drain output without pull-up (Fig. 8c),
- P23 only with open drain configuration.

Serial I/O

The MAB8400 family serial I/O interface has been designed to eliminate the heavy processing load imposed upon a normal microcomputer performing serial data transfer. Whereas a normal microcomputer must regularly monitor the serial data bus for the presence of data, the serial I/O interface detects, receives and converts the serial data stream into parallel format without interrupting the execution of the current program. An interrupt is sent to the microcomputer only when a complete byte is received. Then, the microcomputer reads the data byte in one instruction. Likewise, for transmission, the serial I/O interface performs parallel to serial conversion and subsequent serial output of the data and the microcomputer is only interrupted in the execution of its programmed tasks when a complete byte has been transmitted.

The design of the MAB8400 family serial I/O system allows any number of MAB8400 family devices to be interconnected by the two-line serial bus. The ability of any two devices to communicate, without interrupting the operation of any other devices on the bus, is an outstanding attribute of the system. This is achieved by allocating a specific 7-bit address to each device and providing a system whereby a device reacts only to message prefixed with its own address or the 'general call' address. Address recognition is performed by the interface hardware so that operation of the microcomputer need only be interrupted when a valid address has been received. This saves significant processing time and memory space compared with a convential microcomputer employing a software serial interface. When the addressing facility is not required, for instance in a system with only two microcomputers, direct data transfer without addressing can be performed. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices from continuing simultaneous transmission.

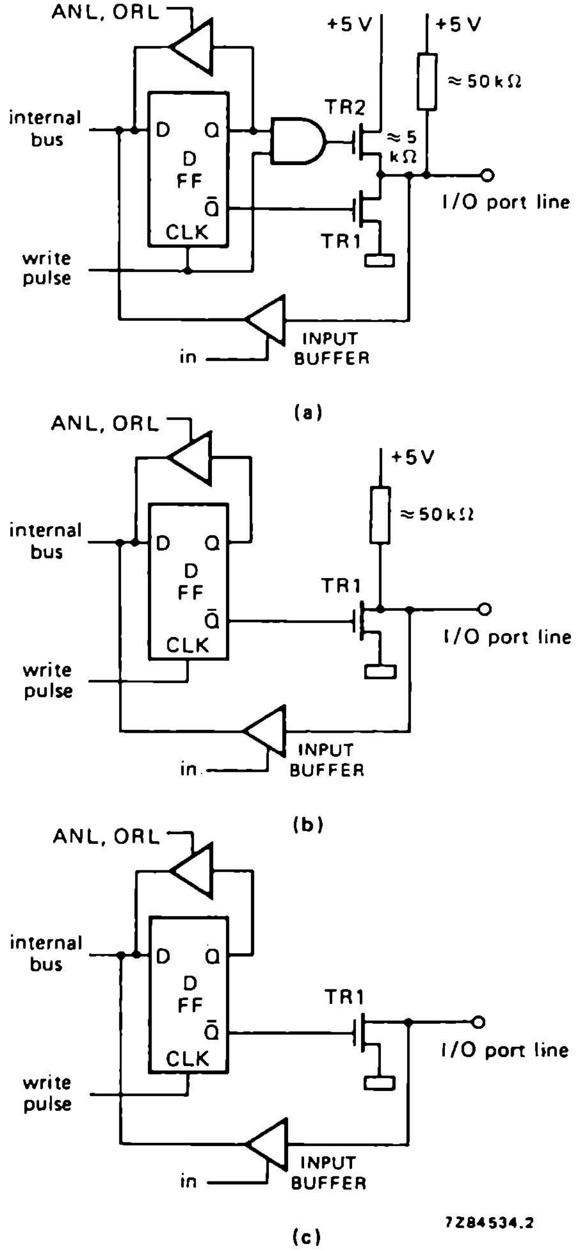

ANL, ORL
internal bus
D
Q
D
FF
Q̄
CLK
write pulse
+5 V
+5 V
≈ 50 kΩ
TR2
≈ 5 kΩ
TR1
I/O port line
INPUT BUFFER
in
(a)
ANL, ORL
internal bus
D
Q
D
FF
Q̄
CLK
write pulse
+5V
≈ 50 kΩ
TR1
I/O port line
INPUT BUFFER
in
(b)
ANL, ORL
internal bus
D
Q
D
FF
Q̄
CLK
write pulse
TR1
I/O port line
INPUT BUFFER
in
7Z84534.2
(c)

Fig. 8 Quasi-bidirectional I/O interface with (a) push-pull output with pull-up resistance; (b) open drain output with pull-up resistance; (c) open drain output without pull-up resistance..

FUNCTIONAL DESCRIPTION (continued)

Serial I/O interface

Figure 9 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 3 (SCLK) while the data line shares pin 2 (serial data) with the I/O line P23 of port 2. When the serial I/O is enabled, P23 is disabled as a parallel port line; (P23 and SCLK only open drain).

The microcomputer and interface communicate via the internal microcomputer bus and the Serial Interrupt Request line. Data and information controlling the operation of the interface are stored in four registers:
- data shift register S0,
- serial I/O interface status word S1,
- serial clock control word S2,
- address register.

Data shift register.

S0 is the shift register that converts serial data to parallel format and vice versa. A pending interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific address or general call address has been received. The most significant bit is transmitted first.

Status word S1.

S1 provides information about the state of the interface and stores interface control information from the microcomputer. Bits 0 to 3 are duplicated: control bits in these positions can only be written by the microcomputer, while interface status bits can only be read.

MST and TRX

These bits determine the operating mode of the serial I/O interface (Table 1).

Table 1 Operating modes of the serial I/O interface

MST	TRX	mode
0	0	slave receiver
1	0	master receiver
0	1	slave transmitter
1	1	master transmitter

BB: Bus Busy

This is the flag which indicates the status of the bus.

PIN: Pending Interrupt Not

PIN = '0' indicates the presence of a pending interrupt, which will cause a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables, ESO = '0' disables.

BC0, BC1 and BC2

These bits indicate the number of bits received or transmitted in a serial data stream.

Bits ESO, BC0, BC1 and BC2 can only be written by software.

AL: Arbitration Lost

The arbitration lost flag is set by hardware when the serial I/O interface, as master transmitter, loses a bus arbitration procedure.

AAS: Addressed As Slave

This flag is set by hardware when the interface detects either its own specific address or the general call address as the first byte of a transfer and the interface has been programmed to operate in the • address recognition mode.

AD0: Address Zero

This flag is set by hardware after detection of the general call address when the interface is operating in the address recognition mode.

LRB: Last Received Bit

This contains either the last data bit received or, for a transmitting device in the acknowledgement mode, the acknowledgement signal from the receiving device.

Bits AL, AAS, AD0 and LRB can only be read by software.

Clock control register S2

Bits 0 to 4 of S2 are used to set the frequency of the serial clock signal. When a 4,43 MHz crystal is used, the frequency of the serial clock can be varied between 100 kHz and 720 Hz. An asymmetrical clock with a HIGH to LOW ratio of 3 : 1 can be generated using bit 5. The asymmetrical clock allows a microcomputer more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 can be used to activate the acknowledge mode of the serial I/O.
S2 is a write-only register.

Address register

The address register contains the 7-bit address back-up latches and the bit (ALS) used to enable/disable the address recognition mode. The address register can be written using the MOV S0, A and MOV S0, ≠ data instructions, but only when ESO = '0'.

Serial I/O interrupt logic

An EN SI instruction enables and a DIS SI instruction disables the interrupt logic. When the logic is enabled, a pending interrupt results in a serial I/O interrupt to the processor, causing a call to location 5 in the ROM. When disabled, the presence of an interrupt is still indicated by PIN in S1, thus the interrupt can still be serviced. However, vectored interrupt will not occur.

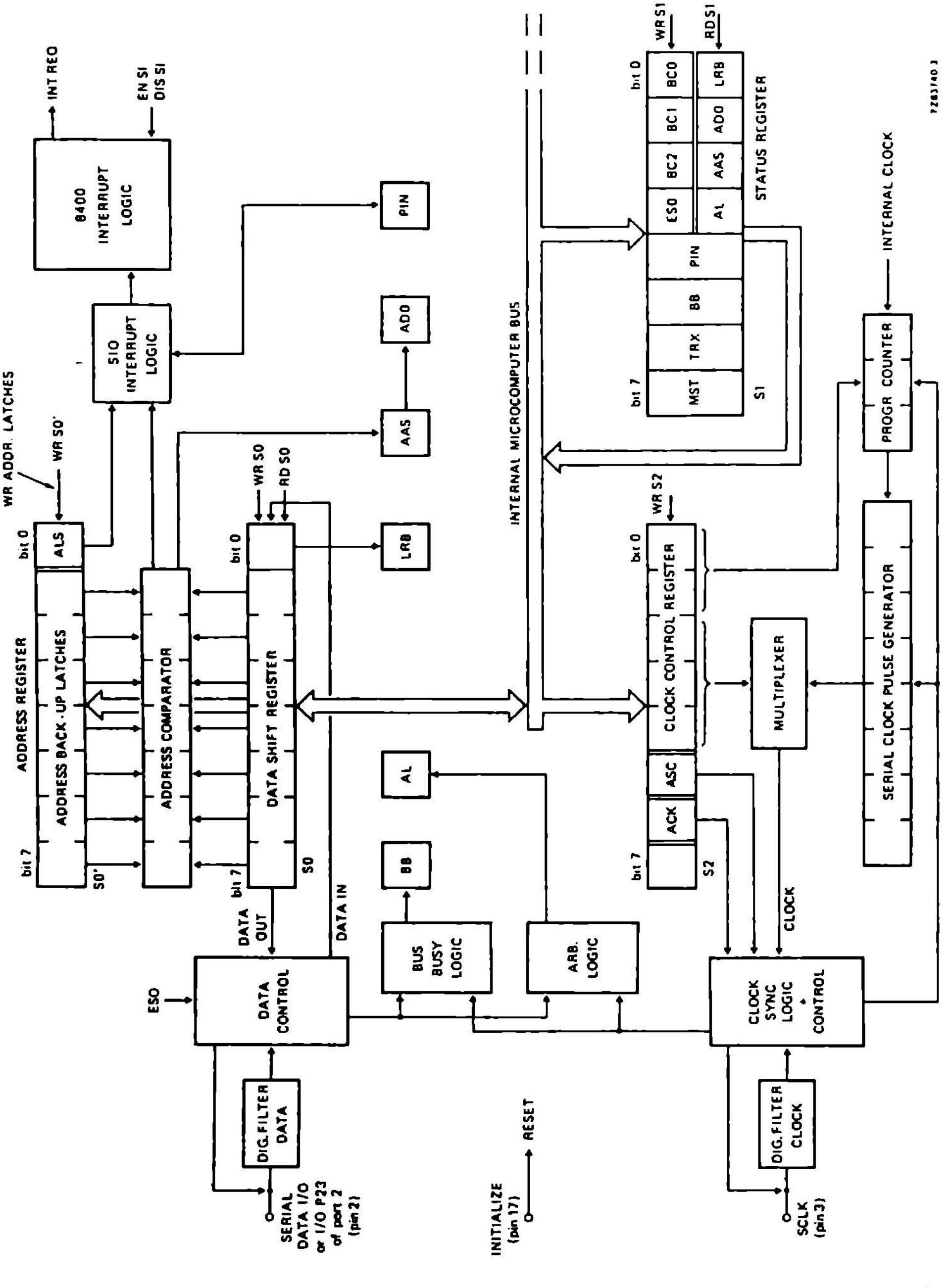

INT REQ
EN SI
DIS SI
8400 INTERRUPT LOGIC
SIO INTERRUPT LOGIC
PIN
ADO
AAS
LRB
AL
BB
WR ADDR. LATCHES
WR S0'
bit 0
ALS
ADDRESS REGISTER
ADDRESS BACK-UP LATCHES
bit 7
S0'
ADDRESS COMPARATOR
WR S0
RD S0
DATA SHIFT REGISTER
S0
DATA OUT
DATA IN
ESO
DATA CONTROL
DIG. FILTER DATA
SERIAL DATA I/O or I/O P23 of port 2 (pin 2)
BUS BUSY LOGIC
ARB. LOGIC
INITIALIZE (pin 17)
RESET
INTERNAL MICROCOMPUTER BUS
WR S1
RD S1
bit 0
BC0
BC1
BC2
ESO
LRB
ADO
AAS
AL
PIN
BB
TRX
MST
bit 7
S1
STATUS REGISTER
WR S2
CLOCK CONTROL REGISTER
ASC
ACK
S2
MULTIPLEXER
CLOCK
CLOCK SYNC LOGIC + CONTROL
DIG. FILTER CLOCK
SCLK (pin 3)
INTERNAL CLOCK
PROGR. COUNTER
SERIAL CLOCK PULSE GENERATOR
7Z83740.3

Fig. 9 Serial I/O interface.

FUNCTIONAL DESCRIPTION (continued)

Interrupt

When the external interrupt is enabled, a HIGH to LOW transition on the $\overline{\text{INT}}$/T0 input initiates an external interrupt subroutine which causes a call to program memory location 3 following completion of the current instruction. The interrupt must remain enabled until the interrupt instruction is completed, otherwise the next instruction of the main program will be executed. Serial I/O interrupt, when enabled, causes a call to location 5, and a timer/event counter overflow a call to location 7, when the interrupt is enabled.

When the external interrupt is disabled, an external interrupt is latched. Therefore, keyboard or sensor interrupt requests are not lost when the processor must first perform some necessary functions whilst the external interrupt is disabled. When an interrupt subroutine starts, the program counter contents and bits 4, 6 and 7 of the PSW have been saved in the program counter stack. Accumulator contents have to be saved by software. Interrupt acknowledgement can be carried out by software via port pins. All interrupt routines must reside in memory bank 0.

The interrupt system is single-level — once an interrupt is detected, further interrupt requests are latched but ignored until the execution of a RETR instruction re-enables the interrupt input logic. After executing RETR, the program continues in the main part; this is independent of the occurrence of a second interrupt during the running of the first routine. If 2 or all 3 interrupts occur simultaneously, their priority is: (1) external, (2) serial I/O, (3) timer/event counter.

Another external interrupt can be created by enabling the timer/event counter interrupt loading FFH into the counter (one less than overflow) and enabling the event counter mode. A LOW to HIGH transition on the T1 input will then initiate an interrupt subroutine and cause a call to location 7.

Test input T1

The T1 input line can be used as:
- a test input for branch instructions,
- an input for zero voltage cross-over detection,
- an external input to the event counter.

A pull-up resistor can be provided as a ROM mask option. This is useful when the input is from a switch or a standard TTL output.

When T1 is used as a test input, the JT1 and JNT1 instructions test for 1 and 0 levels respectively. The T1 input has a self-biasing circuit which can detect when an a.c. signal crosses zero (within ± 100 mV when coupled through a 1 μF capacitor). The maximum input voltage is 3 V (peak-to-peak), the maximum frequency 1 kHz. Zero cross-over detection used in conjunction with the timer/event counter interrupt is useful in thyristor control of power equipment.

The operation of T1 as an input to the event counter is described under the heading *Timer/event counter.*

High current outputs

Four pins are provided which can sink higher currents (typical values):

– P23 (serial data, pin 2	5 mA at 0,45 V (open drain),
– SCLK, pin 3	5 mA at 0,45 V (open drain),
– P10, pin 18	7 mA at 2,5 V,
– P11, pin 19	7 mA at 2,5 V.

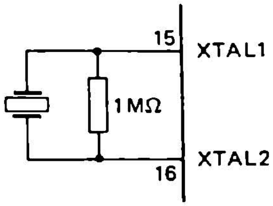
P10 and P11 may be paralled (if their logic outputs are always the same) to give 14 mA drive at 2,5 V.

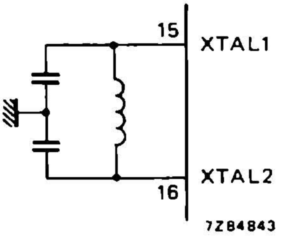
FUNCTIONAL DESCRIPTION (continued)

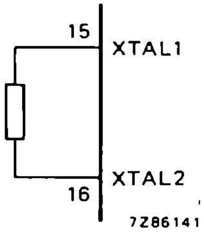
Oscillator and clock

A crystal, inductor or resistor connected between XTAL1 and XTAL2 (see below) determines the frequency of the internal oscillator. An externally generated clock signal can also be applied to XTAL1 as the frequency reference. A machine cycle consists of 10 states, each state being 3 oscillator periods. The common 4,43 MHz crystal gives a 6,77 μs machine cycle.

The MAB8400 family has dynamic logic; for adequate refreshing the oscillator frequency must be > 600 kHz.







Timer/event counter

An internal 8-bit binary up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly (Fig. 10). Table 2 gives the instructions that control the counter and the prescaler and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW to HIGH transitions on T1 (pin 13) are counted. The maximum rate at which the counter may be incremented is once every machine cycle (147,7 kHz for a 6,77 μs machine cycle). When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump if timer flag = 1) instruction or JNTF instruction. Overflow also generates an interrupt to the processor when the timer/event counter interrupt is enabled.

Table 2 Timer/event counter control

function	timer mode modulo-1, module-32*	counter mode
CLEAR	MOV T,A (A) = 0 or RESET	MOV T,A (A) = 0 or RESET
PRESET	MOV T,A	MOV T,A
START	STRT T	STRT CNT
STOP	STOP TCNT or RESET	STOP TCNT or RESET
TEST	JTF/JNTF	JTF/JNTF
READ**	MOV A,T	MOV A,T

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

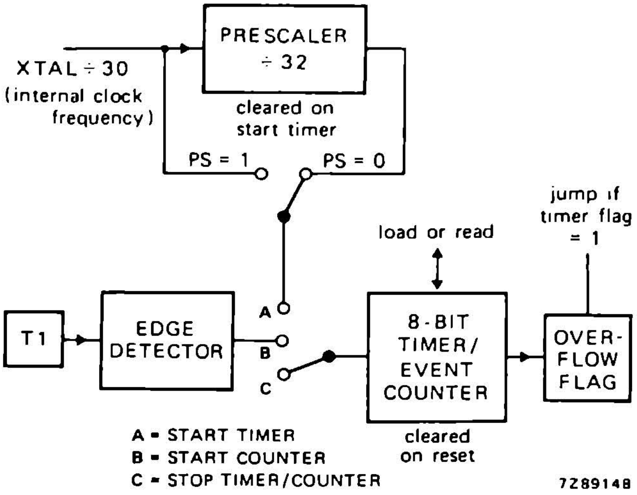

PRESCALER
÷ 32
XTAL ÷ 30
(internal clock
frequency)
cleared on
start timer
PS = 1
PS = 0
load or read
jump if
timer flag
= 1
T1
EDGE
DETECTOR
A
B
C
8-BIT
TIMER/
EVENT
COUNTER
OVER-
FLOW
FLAG
A = START TIMER
B = START COUNTER
C = STOP TIMER/COUNTER
cleared
on reset
7Z89148

Fig. 10 Timer/event counter.

Program status word

The program status word (PSW) is an 8-bit word (1-byte) in the CPU which stores information about the current status of the microcomputer (Fig. 11). The PSW bits are:

bits 0, 1 and 2 – stack pointer bits (SP_0, SP_1, SP_2),
bit 3 – prescaler select (PS); 0 = modulo-32; 1 = modulo-1 (no prescaling),
bit 4 – working register bank select (RBS); 0 = register bank 0; 1 = register bank 1,
bit 5 – not used (1),
bit 6 – auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A,
bit 7 – carry (CY); the carry flag indicates that the previous operation has resulted in an overflow of the accumulator.

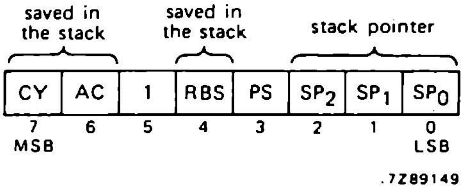

saved in
the stack
saved in
the stack
stack pointer
CY
AC
1
RBS
PS
SP2
SP1
SP0
7
6
5
4
3
2
1
0
MSB
LSB
7Z89149

Fig. 11 Program status word.

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in case of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

FUNCTIONAL DESCRIPTION (continued)

Program counter

A 13-bit program counter is used so that up to 8K bytes of ROM can be addressed. Figure 12 shows the arrangement of the bits. During an interrupt subroutine PC_{11} and PC_{12} are forced to 0. All 13 bits are saved in the stack during CALL and interrupt routines.

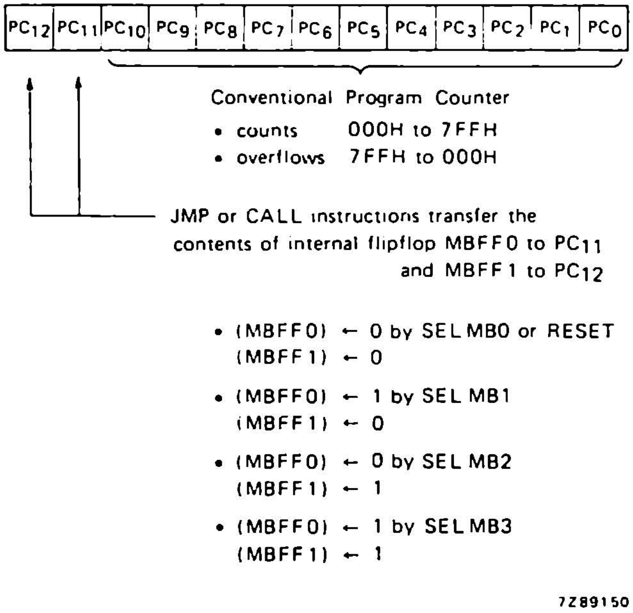

PC12 PC11 PC10 PC9 PC8 PC7 PC6 PC5 PC4 PC3 PC2 PC1 PC0
Conventional Program Counter
• counts 000H to 7FFH
• overflows 7FFH to 000H
JMP or CALL instructions transfer the contents of internal flipflop MBFF0 to PC11 and MBFF1 to PC12
• (MBFF0) ← 0 by SEL MB0 or RESET
(MBFF1) ← 0
• (MBFF0) ← 1 by SEL MB1
(MBFF1) ← 0
• (MBFF0) ← 0 by SEL MB2
(MBFF1) ← 1
• (MBFF0) ← 1 by SEL MB3
(MBFF1) ← 1
7Z89150

Fig. 12 Program counter.

Central processing unit

The MAB8400 family has arithmetic, logical and branching capabilities. The DA A, SWAP A, and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look up from the current ROM page.

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 3 lists the conditional jump instructions used to change the program execution sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

Table 3 Conditional branches

test	jump condition	jump instruction
accumulator	0 or non-zero	JZ, JNZ
accumulator bit test	1	JB0 to JB7
carry flag	0 or 1	JNC, JC
timer overflow flag	0 or 1	JNTF, JTF
test input T0	0 or 1	JNT0, JT0
test input T1	0 or 1	JNT1, JT1
register	non-zero	DJNZ

Reset

A positive-going signal on the RESET input:
- sets the program counter to zero,
- selects location 0 of memory bank 0, and register bank 0,
- sets the stack pointer to zero (000); pointing to RAM address 8,
- disables the interrupts (external, timer and serial I/O),
- stops the timer/event counter, then sets it to zero,
- sets the timer prescaler to modulo-32,
- resets the timer flag,
- sets all ports to logic '1' (input mode),
- sets the serial I/O to slave receiver mode and disables the serial I/O.

The external power-on-reset circuit can consist of a capacitor connected between V_{CC} and the RESET pin. A diode may be added between the RESET pin and ground to endure reset if the supply voltage falls momentarily.
RESET has to be active HIGH for at least > 2 machine cycles after the power supply and clock have stabilized.

TABLE 4. MAB8400 INSTRUCTION MAP

first hexadecimal character of opcode

second hexadecimal character of opcode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP			ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp 0	1	2		MOV A,Sn 0	1		
1	INC @Rr 0	1	JB0 addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr 0	1	2	3	4	5	6	7
2	XCH A,@Rr 0	1		MOV A,#data	JMP page 1	EN TCNTI	JNT0 addr	CLR A	XCH A,Rr 0	1	2	3	4	5	6	7
3	XCHD A,@Rr 0	1	JB1 addr		CALL page 1	DIS TCNTI	JT0 addr	CPL A	OUTL Pp,A 0	1	2		MOV Sn,A 0	1	2	
4	ORL A,@Rr 0	1	MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr 0	1	2	3	4	5	6	7
5	ANL A,@Rr 0	1	JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr 0	1	2	3	4	5	6	7
6	ADD A,@Rr 0	1	MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr 0	1	2	3	4	5	6	7
7	ADDC A,@Rr 0	1	JB3 addr		CALL page 3			RR A	ADDC A,Rr 0	1	2	3	4	5	6	7
8				RET	JMP page 4	EN SI			ORL Pp,#data 0	1	2					
9			JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data 0	1	2		MOV Sn,#data 0	1	2	
A	MOV @Rr,A 0	1		MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A 0	1	2	3	4	5	6	7
B	MOV @Rr,#data 0	1	JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data 0	1	2	3	4	5	6	7
C	DEC @Rr 0	1			JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr 0	1	2	3	4	5	6	7
D	XRL A,@Rr 0	1	JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr 0	1	2	3	4	5	6	7
E	DJNZ @Rr,addr 0	1			JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr 0	1	2	3	4	5	6	7
F	MOV A,@Rr 0	1	JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr 0	1	2	3	4	5	6	7

Instruction set

The MAB8400 family instruction set consists of over 80 one and two byte instructions and is based on the MAB8048 instruction set. New instructions include those for serial I/O operation and memory bank selection. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Table 7 gives the instruction set of the MAB8400 family; Table 4 shows the instruction map. The following symbols and abbreviations are used:

symbol	description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0–7)
RBS	register bank select
C	carry (bit CY)
CNT	event counter
D	mnemonic for 4-bit digit (nibble)
data	8-bit number or expression
I	interrupt
MB	memory bank
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1, 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0–7)
S_n	serial I/O register
SP	stack pointer
T	timer
TF	timer flag
T1	test 1 input
T0	test 0 input
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

FUNCTIONAL DESCRIPTION (continued)

Table 5 shows the MAB8400 family instructions (including the five instructions for serial I/O operation) absent from the MAB8048 instruction set.

Table 5 MAB8400 family instructions not in the MAB8048 instruction set

serial I/O	register	control	conditional branch

MOV A, S_n
MOV S_n, A
MOV S_n, #data
EN SI
DIS SI
	DEC @Rr
	DJNZ @Rr, addr
		SEL MB2
		SEL MB3
			JNTF addr

Table 6 shows the MAB8048 instructions absent from the MAB8400 family instruction set.

Table 6 MAB8048 instructions not in the MAB8400 family instruction set

data moves	flags	branch	control

MOVX A, @R
MOVX @R, A
MOVP3 A, @A
MOVD A, P
MOVD P, A
ANLD P, A
ORLD P, A
	CLR F0
	CPL F0
	CLR F1
	CPL F1
		* JNI addr
		JF0 addr
		JF1 addr
		* replaced by JT0, JNT0.
			ENT0 CLK

Differences between the MAB8021, MAB8048 microcomputers and the MAB8400 family:

	8021	8048	8410, 8420, 8440, 8400
ROM capacity (bytes)	1K	1K	1K, 2K, 4K, ROMless
RAM capacity (bytes)	64	64	64, 64, 128, 128
parallel I/O lines	8 + 8 + 4	8 + 8 + 8	8 + 8 + 4
single inputs	1	3	2
serial I/O	no	no	yes, 2-line multi-transmitter
timer	8 bit	8 bit	8 bit
prescaler	mod. 32	mod. 32	mod. 1 & mod. 32
machine cycle time (μs)	10	2,5	6,7
for clock (MHz)	3	6	4,43
instruction set	8021	8048	8048 with omissions; 5 new serial I/O instructions; 2 new register instructions; 2 new control instructions; 1 new cond. branch instruction
interrupts	none	2 external timer/ event counter	3 external serial I/O timer/event counter
no. of pins (DIL)	28	40	28

Table 7 Instruction set is shown on the next 5 pages.

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr) r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))	1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data	1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C) r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)	1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)	1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr) r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))	
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data	
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr) r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))	
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data	
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr) r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))	
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data	
	INC A	17	1/1	increment A by 1	(A)←(A) + 1	
	DEC A	07	1/1	decrement A by 1	(A)←(A) – 1	
	CLR A	27	1/1	clear A to zero	(A)←0	
	CPL A	37	1/1	one's complement A	(A)←NOT(A)	
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ n = 0–6 $(A_0)\leftarrow(A_7)$	

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1}) \leftarrow A_n$ n = 0–6 $(A_0) \leftarrow (C)$, $(C) \leftarrow (A_7)$	2
	RR A	77	1/1	rotate A right	$(A_n) \leftarrow (A_{n+1})$ n = 0–6 $(A_7) \leftarrow (A_0)$	
	RRC A	67	1/1	rotate A right through carry	$(A_n) \leftarrow (A_{n+1})$ n = 0–6 $(A_7) \leftarrow (C)$, $(C) \leftarrow (A_0)$	2
	DA A	57	1/1	decimal adjust A		2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7}) \leftrightarrow (A_{0-3})$	
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	$(A) \leftarrow (Rr)$ r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	$(A) \leftarrow ((R0))$ $(A) \leftarrow ((R1))$	
	MOV A, #data	23 data	2/2	move immediate data to A	$(A) \leftarrow$ data	
	MOV Rr, A	A*	1/1	move accumulator contents to register	$(Rr) \leftarrow (A)$ r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	$((R0)) \leftarrow (A)$ $((R1)) \leftarrow (A)$	
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	$(Rr) \leftarrow$ data	
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	$((R0)) \leftarrow$ data $((R1)) \leftarrow$ data	
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	$(A) \leftrightarrow (Rr)$ r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	$(A) \leftrightarrow ((R0))$ $(A) \leftrightarrow ((R1))$	
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3}) \leftrightarrow ((R0_{0-3}))$ $(A_{0-3}) \leftrightarrow ((R1_{0-3}))$	
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	$(A) \leftarrow (PSW)$	
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3) \leftarrow (A_3)$	3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7}) \leftarrow (A)$, $(A) \leftarrow ((PC))$	

FLAGS	CLR C	97	1/1	clear carry bit	$(C)\leftarrow 0$		2
	CPL C	A7	1/1	complement carry bit	$(C)\leftarrow NOT(C)$		2
REGISTER	INC Rr	1*	1/1	increment register by 1	$(Rr)\leftarrow(Rr)+1$	r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	$((R0))\leftarrow((R0))+1$ $((R1))\leftarrow((R1))+1$		
	DEC Rr	C*	1/1	decrement register by 1	$(Rr)\leftarrow(Rr)-1$	r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	$((R0))\leftarrow((R0))-1$ $((R1))\leftarrow((R1))-1$		
BRANCH	JMP addr	●4 address	2/2	unconditional jump within a 2Kbank	$(PC_{8-10})\leftarrow addr_{8-10}$ $(PC_{0-7})\leftarrow addr_{0-7}$ $(PC_{11-12})\leftarrow$MBFF 0–1		
	JMPP @A	B3	1/2	indirect jump within a page	$(PC_{0-7})\leftarrow((A))$		
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	$(Rr)\leftarrow(Rr)-1$ if (Rr) not zero $(PC_{0-7})\leftarrow$addr	r = 0–7	
	DJNZ @Rr, addr	E0	2/2	decrement RAM data, addressed by Rr, by 1 and jump if not zero to addr	$((R0))\leftarrow((R0))-1$ if ((R0)) not zero $(PC_{0-7})\leftarrow$addr		
		E1			$((R1))\leftarrow((R1))-1$ if ((R1)) not zero $(PC_{0-7})\leftarrow$addr		
	JBb addr	▲2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1: $(PC_{0-7})\leftarrow$addr	b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1: $(PC_{0-7})\leftarrow$addr		
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0: $(PC_{0-7})\leftarrow$addr		
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0. $(PC_{0-7})\leftarrow$addr		
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A ≠ 0: $(PC_{0-7})\leftarrow$addr		
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: $(PC_{0-7})\leftarrow$addr		
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0· $(PC_{0-7})\leftarrow$addr		
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: $(PC_{0-7})\leftarrow$addr		
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: $(PC_{0-7})\leftarrow$addr		
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: $(PC_{0-7})\leftarrow$addr		4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: $(PC_{0-7})\leftarrow$addr		

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
SUBROUTINE	CALL addr	▲4 address	2/2	jump to subroutine	((SP))←(PC), $(PSW_{4,6,7})$ (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) – 1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) – 1 $(PSW_{4,6,7})$ + (PC)←((SP))	6

PARALLEL INPUT/OUTPUT	IN A, Pp	08	1/2	input port p data to accumulator	(A)←(P0)	
		09			(A)←(P1)	
		0A			(A)←(P2)	7
	OUTL Pp, A	38	1/2	output accumulator data to port p	(P0)←(A)	
		39			(P1)←(A)	
		3A			(P2)←(A)	
	ANL Pp, #data	98	2/2	AND port p data with immediate data	(P0)←(P0) AND data	
		99			(P1)←(P1) AND data	
		9A			(P2)←(P2) AND data	
	ORL Pp, #data	88	2/2	OR port p data with immediate data	(P0)←(P0) OR data	
		89			(P1)←(P1) OR data	
		8A			(P2)←(P2) OR data	
SERIAL INPUT/OUTPUT	MOV A, S_n	0C	1/2	move serial I/O register contents to accumulator	(A)←(S0)	
		0D			(A)←(S1)	8
	MOV S_n, A	3C	1/2	move accumulator contents to serial I/O register	(S0)←(A)	
		3D			(S1)←(A)	
		3E			(S2)←(A)	
	MOV S_n, #data	9C	2/2	move immediate date to serial I/O register	(S0)←data	
		9D			(S1)←data	
		9E			(S2)←data	
	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 6.

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 1111 P23, P22, P21, P20.
8. (S1) has a different meaning for read and write operation, see serial I/O interface.

· : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage with respect to V_{SS}	V_{CC}	−0,5 to + 7 V
All input and output voltages with respect to V_{SS} (except for the port pins)	V_I, V_O	−0,5 to + 7 V
Voltage at any port pin with respect to V_{SS}; with R ⩾ 1 kΩ in series	V_I	−0,5 to + 12 V
Total power dissipation		
for SOT-117D	P_{tot}	max. 1 W
for SOT-135 and SOT-136A (SO-28)	P_{tot}	max. 0,6 W
Input output current	$\pm I_I$, I_O	max. 10 mA
Storage temperature range	T_{stg}	−65 to + 150 °C
Operating ambient temperature range	T_{amb}	0 to + 70 °C

D.C. CHARACTERISTICS

V_{SS} = 0 V; V_{CC} = 5 V (± 10%); T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

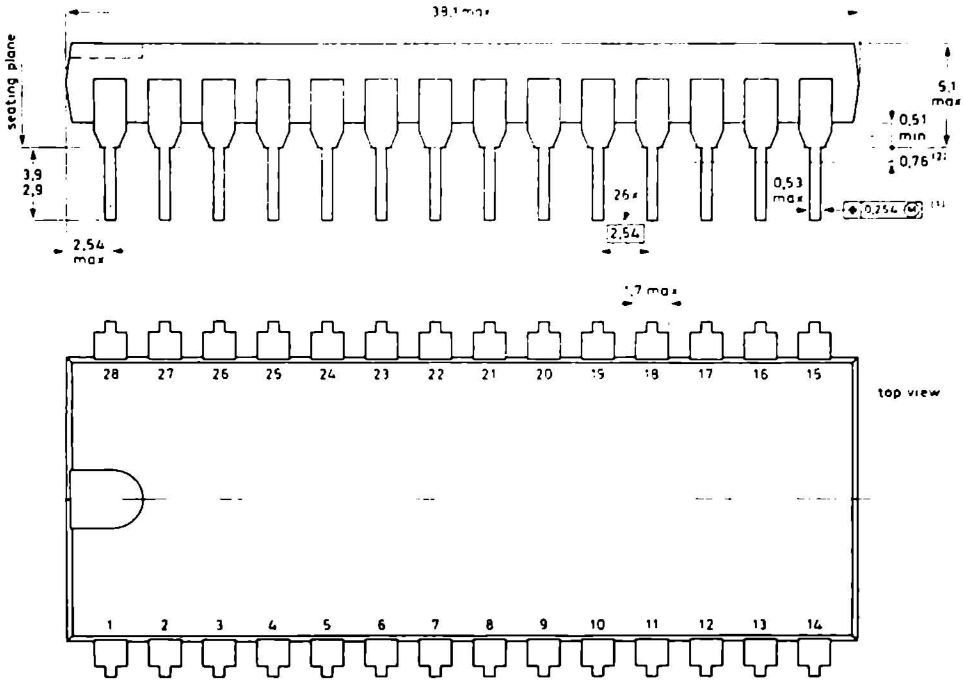
	symbol	min.	typ.	max.		conditions
Supply voltage	V_{CC}	4,5	5,0	5,5	V	
Inputs						
Input voltage LOW all inputs except P23, SCLK	V_{IL}	−0,5	–	0,8	V	
Input voltage HIGH all inputs except XTAL1, P23, SCLK	V_{IH}	2,0	–	V_{CC}	V	
Input voltage HIGH XTAL1, P23, SCLK	V_{IH}	3,0	–	V_{CC}	V	
Input voltage LOW P23, SCLK	V_{IL}	−0,5	–	1,5	V	
Outputs						
Output voltage LOW	V_{OL}	–	–	0,45	V	I_{OL} = 1,6 mA
Output voltage LOW P10, P11	V_{OL}	–	–	2,5	V	I_{OL} = 7 mA
Output voltage LOW P23, SCLK	V_{OL}	–	–	0,45	V	I_{OL} = 5 mA
Output voltage HIGH all outputs unless open drain	V_{OH}	2,4	–	–	V	$-I_{OH}$ = 50 μA
Output leakage current open drain	$-I_{OL}$	–	–	10	μA	$V_{CC} \geqslant V_I \geqslant V_{SS}$

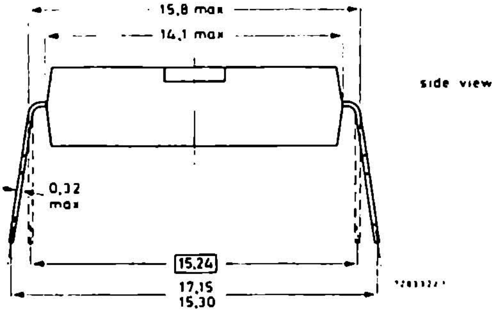
A.C. CHARACTERISTICS

V_{SS} = 0 V; V_{CC} = 5 V (± 10%); T_{amb} = 0 to + 70 °C; unless otherwise specified

	symbol	min.	typ.	max.		condition
Cycle time	t_{cy}	6,77	–	50	μs	4,43 MHz crystal = 6,77 μs

28-LEAD DUAL IN-LINE; PLASTIC (SOT-117D)


38,1 max
seating plane
5,1 max
0,51 min
0,76 (2)
3,9
2,9
0,53 max
26×
2,54
0,254 M (1)
2,54 max
1,7 max
28
27
26
25
24
23
22
21
20
19
18
17
16
15
top view
1
2
3
4
5
6
7
8
9
10
11
12
13
14


15,8 max
14,1 max
side view
0,32 max
15,24
17,15
15,30

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

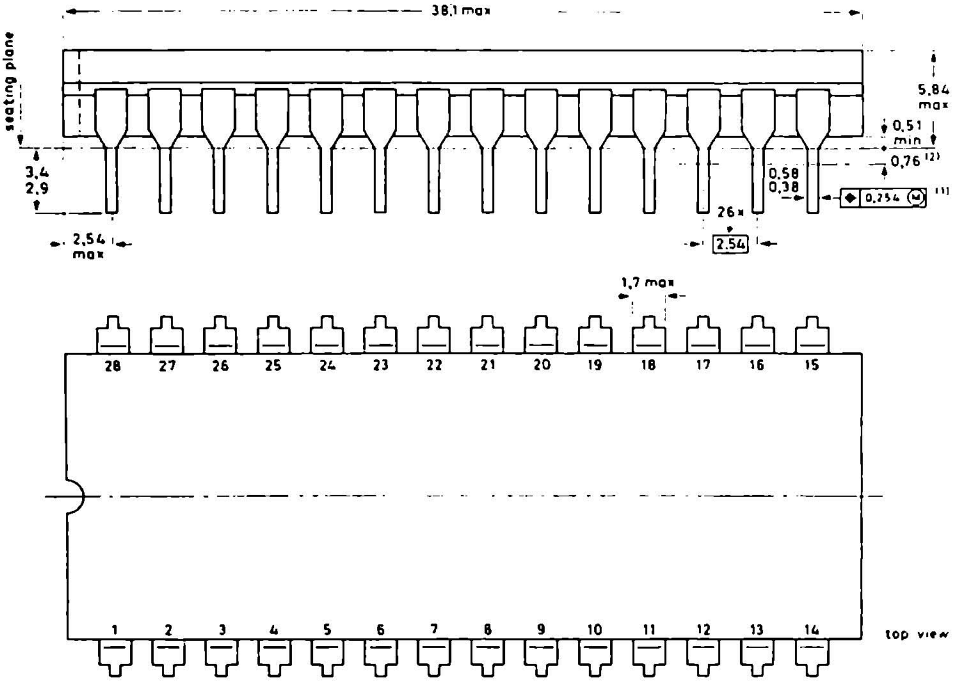
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

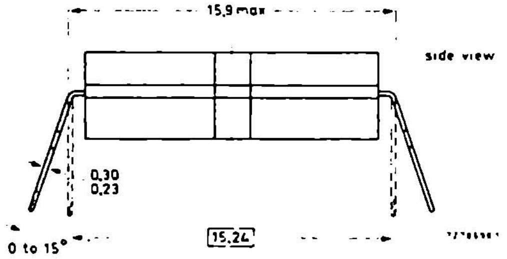
(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

Dimensions in mm

28-LEAD DUAL IN-LINE; CERAMIC (SOT-135)


38,1 max
seating plane
5,84 max
0,51 min
0,76 (2)
3,4
2,9
0,58
0,38
0,254 (M) (1)
26×
2,54
2,54 max
1,7 max
28
27
26
25
24
23
22
21
20
19
18
17
16
15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
top view


15,9 max
side view
0,30
0,23
0 to 15°
15,24

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

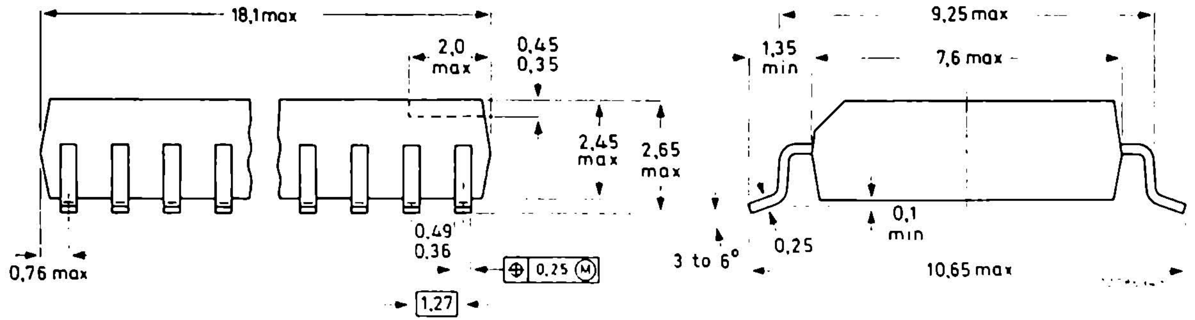
(2) Lead spacing tolerances apply from seating plane to the line indicated.

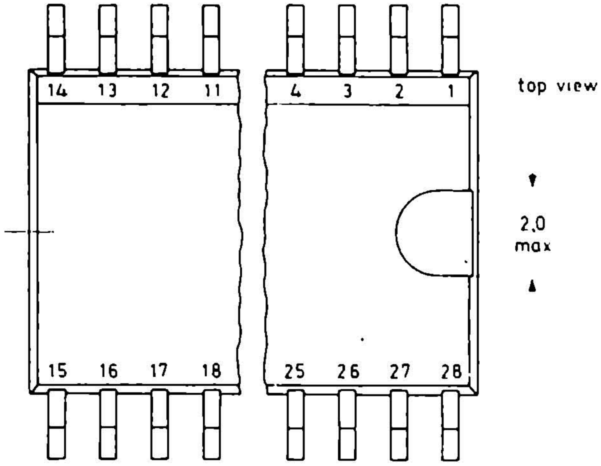
Dimensions in mm

Remarks

1. Leads are given positive misalignment so that they grip after insertion.
2. Leads are Ni-Fe, pure tin plated.

28-LEAD FLAT-PACK; PLASTIC (SO-28; SOT-136A)


18,1max
2,0 max
0,45
0,35
2,45 max
2,65 max
0,49
0,36
0,76 max
0,25 (M)
1,27
9,25 max
1,35 min
7,6 max
0,1 min
0,25
3 to 6°
10,65 max

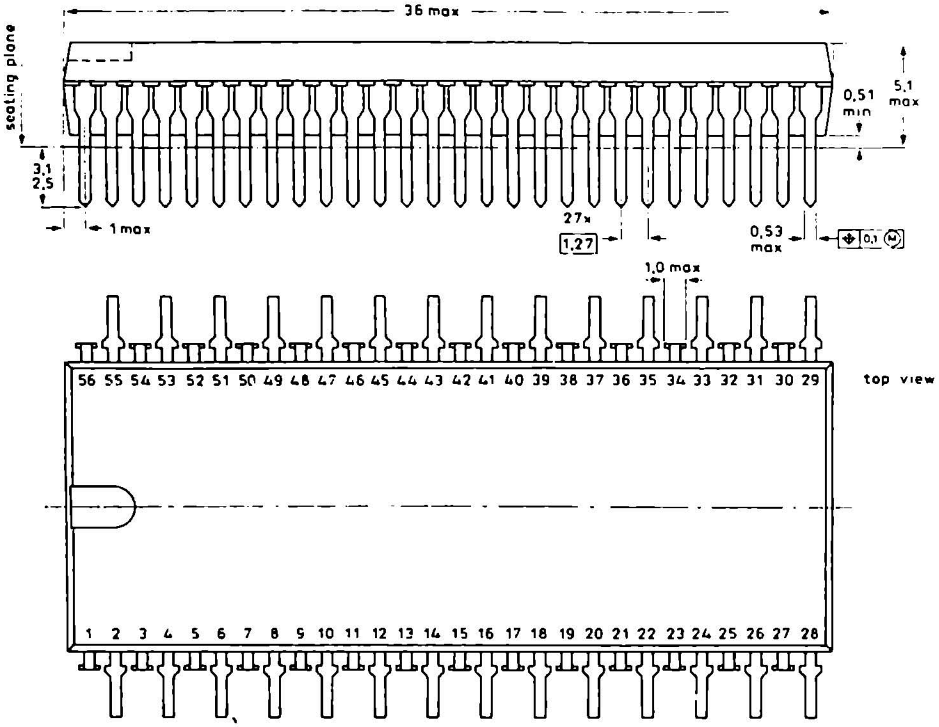

14
13
12
11
4
3
2
1
15
16
17
18
25
26
27
28
top view
2,0 max

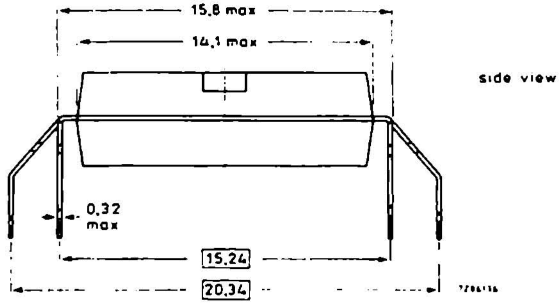
Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

56-LEAD QUADRUPLE IN-LINE; PLASTIC (VO-35)


36 max
seating plane
0,51
min
5,1
max
3,1
2,5
1 max
27×
1,27
0,53
max
0,1 M
1,0 max
56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29
top view
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28


15,8 max
14,1 max
side view
0,32
max
15,24
20,34

Dimensions in mm

⨁ Positional accuracy.

Ⓜ Maximum Material Condition.

Section 4
SC 68 000
16-Bit Microprocessor-Family

VALVO

16-BIT MICROPROCESSOR

SC68000 SERIES

Preliminary

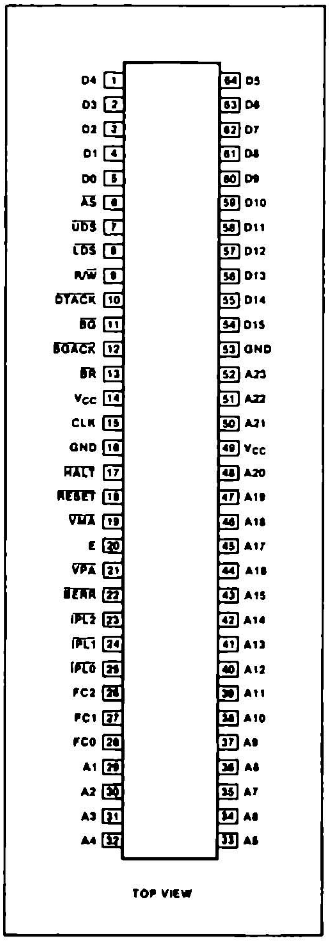
DESCRIPTION

The Signetics SC68000 combines state-of-the-art semiconductor technology and advanced circuit design techniques with computer sciences to achieve an architecturally advanced 16-bit microprocessing unit. As shown in the programming model, figure 1, the SC68000 offers seventeen 32-bit registers in addition to a 32-bit program counter and a 16-bit status register. The first eight registers (D0-D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) data operations. The second set of seven registers (A0-A6) and the system stack pointer can be used as software stack pointers and base address registers. In addition, these registers can be used for word and long word address operations. All 17 registers can be used as index registers.

FEATURES

- **32-bit data and address registers**
- **16 megabyte direct addressing range**
- **56 powerful instruction types**
- **Operations on five main data types**
- **Memory mapped I/O**
- **14 addressing modes**
- **Multilevel interrupt structure**
- **Signed and unsigned multiply and divide**
- **User and supervisor modes**
- **Support for high level languages**
- **Testing and debugging support**

PIN CONFIGURATION[1]


D4 1
D3 2
D2 3
D1 4
D0 5
$\overline{\text{AS}}$ 6
$\overline{\text{UDS}}$ 7
$\overline{\text{LDS}}$ 8
R/$\overline{\text{W}}$ 9
$\overline{\text{DTACK}}$ 10
$\overline{\text{BG}}$ 11
$\overline{\text{BGACK}}$ 12
$\overline{\text{BR}}$ 13
Vcc 14
CLK 15
GND 16
$\overline{\text{HALT}}$ 17
$\overline{\text{RESET}}$ 18
$\overline{\text{VMA}}$ 19
E 20
$\overline{\text{VPA}}$ 21
$\overline{\text{BERR}}$ 22
$\overline{\text{IPL2}}$ 23
$\overline{\text{IPL1}}$ 24
$\overline{\text{IPL0}}$ 25
FC2 26
FC1 27
FC0 28
A1 29
A2 30
A3 31
A4 32
64 D5
63 D6
62 D7
61 D8
60 D9
59 D10
58 D11
57 D12
56 D13
55 D14
54 D15
53 GND
52 A23
51 A22
50 A21
49 Vcc
48 A20
47 A19
46 A18
45 A17
44 A16
43 A15
42 A14
41 A13
40 A12
39 A11
38 A10
37 A9
36 A8
35 A7
34 A6
33 A5
TOP VIEW

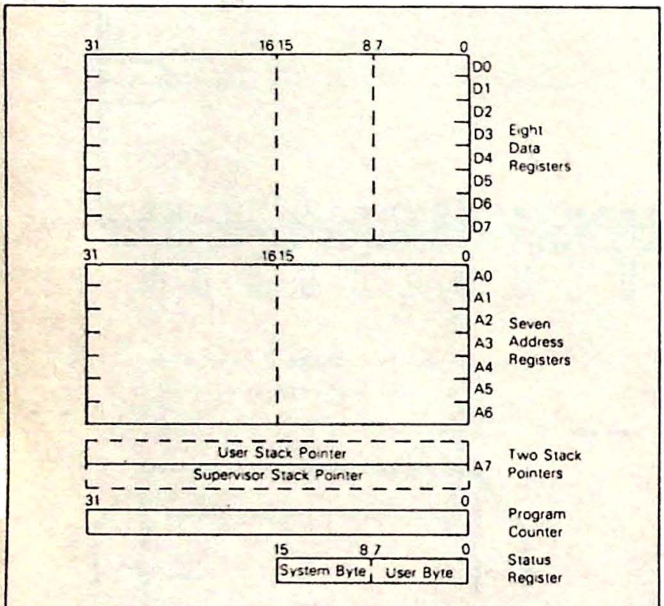

31
16 15
8 7
0
D0
D1
D2
D3
D4
D5
D6
D7
Eight Data Registers
31
16 15
0
A0
A1
A2
A3
A4
A5
A6
Seven Address Registers
User Stack Pointer
Supervisor Stack Pointer
A7
Two Stack Pointers
31
0
Program Counter
15
8 7
0
System Byte
User Byte
Status Register

Figure 1. Programming Model

ORDERING CODE

PACKAGES	$V_{DD} = 5V \pm 5\%, T_A = 0°$ to $70°C$			
	4MHz	6MHz	8MHz	10MHz
Ceramic DIP	SC68000C4I64	SC68000C6I64	SC68000C8I64	SC68000CAI64

---

[1] In this data sheet, barring signal names (overscore) to indicate low is done only for the pin configuration diagram, signal description headings, tables and figures.

16-BIT MICROPROCESSOR

SC68000 SERIES

Preliminary

SIGNAL DESCRIPTION

The input and output signals can be functionally organized into the groups shown in figure 2. The I/O signal characteristics are summarized in table 1.

Address Bus (A1-A23)

This 23-bit, unidirectional, three-state bus is capable of addressing eight megawords of data. It provides the address for bus operation during all cycles except interrupt cycles. During interrupt cycles, address lines A1, A2, and A3 provide information about what level interrupt is being serviced while address lines A4-A23 are all set to a logic high.

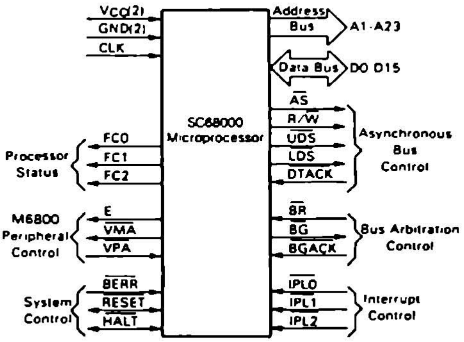

Vcc(2)
GND(2)
CLK
Address Bus
A1-A23
Data Bus
D0-D15
SC68000 Microprocessor
$\overline{\text{AS}}$
R/$\overline{\text{W}}$
$\overline{\text{UDS}}$
$\overline{\text{LDS}}$
$\overline{\text{DTACK}}$
Asynchronous Bus Control
Processor Status
FC0
FC1
FC2
M6800 Peripheral Control
E
$\overline{\text{VMA}}$
$\overline{\text{VPA}}$
$\overline{\text{BR}}$
$\overline{\text{BG}}$
$\overline{\text{BGACK}}$
Bus Arbitration Control
System Control
$\overline{\text{BERR}}$
$\overline{\text{RESET}}$
$\overline{\text{HALT}}$
$\overline{\text{IPL0}}$
$\overline{\text{IPL1}}$
$\overline{\text{IPL2}}$
Interrupt Control

Figure 2. Input and Output Signals

Table 1. SIGNAL SUMMARY

Signal Name	Mnemonic	Input/Output	Active State	Three State
Address Bus	A1-A23	output	high	yes
Data Bus	D0-D15	input/output	high	yes
Address Strobe	$\overline{\text{AS}}$	output	low	yes
Read/Write	R/$\overline{\text{W}}$	output	read-high write-low	yes
Upper and Lower Data Strobes	$\overline{\text{UDS}}$, $\overline{\text{LDS}}$	output	low	yes
Data Transfer Acknowledge	$\overline{\text{DTACK}}$	input	low	—
Bus Request	$\overline{\text{BR}}$	input	low	—
Bus Grant	$\overline{\text{BG}}$	output	low	no
Bus Grant Acknowledge	$\overline{\text{BGACK}}$	input	low	—
Interrupt Priority Level	$\overline{\text{IPL0}}$, $\overline{\text{IPL1}}$, $\overline{\text{IPL2}}$	input	low	—
Bus Error	$\overline{\text{BERR}}$	input	low	—
Reset	$\overline{\text{RESET}}$	input/output	low	no*
Halt	$\overline{\text{HALT}}$	input/output	low	no*
Enable	E	output	high	—
Valid Memory Address	$\overline{\text{VMA}}$	output	low	yes
Valid Peripheral Address	$\overline{\text{VPA}}$	input	low	—
Function Code Output	FC0, FC1, FC2	output	high	yes
Clock	CLK	input	high	no
Power Input	V_{CC}	input	–	—
Ground	GND	input	–	—

*open drain

Data Bus (D0-D15)

This 16-bit, bidirectional, three state bus is the general purpose data path. It can transfer and accept data in either word or byte length. During an interrupt acknowledge cycle, an external device supplies the interrupt vector on data lines D0-D7.

Asynchronous Bus Control

Asynchronous data transfers are handled using the following control signals:

Address Strobe ($\overline{\text{AS}}$) — This signal indicates that there is a valid address on the address bus.

Read/Write (R/$\overline{\text{W}}$) — This signal defines the data bus transfer as a read or write cycle. The R/W signal also works in conjunction with the upper and lower data strobes as explained in the next paragraph.

Upper and Lower Data Strobes ($\overline{\text{UDS}}$, $\overline{\text{LDS}}$) — These signals control the data on the bus as shown in table 2. When the R/W line is high, the processor will read from the data bus as indicated. When the R/W line is low, the processor will write to the data bus as shown.

Data Transfer Acknowledge ($\overline{\text{DTACK}}$) — This input indicates that the data transfer is completed. When the processor recognizes DTACK during a read cycle, data is latched and the bus cycle is terminated. When DTACK is recognized during a write cycle, the bus cycle is terminated. An active transition of DTACK indicates the termination of a data transfer on the bus.

If the system must run at a maximum rate determined by RAM access times, the relationship between the times at which DTACK and data are sampled is important. All control and data lines are sampled during the SC68000's clock high time. The clock is internally buffered, which results in some slight differences in the sampling and recognition of various signals. The DTACK signal, like other control signals, is internally synchronized to allow for valid operation in an asynchronous system. If the required setup time (#47)[1] is met during S4, DTACK will be recognized during

---

[1] Number references (#) can be found in the AC Electrical Characteristics section and corresponding timing diagrams located towards the end of this data sheet.

Table 2. DATA STROBE CONTROL OF DATA BUS

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8-D15	D0-D7
High	High	–	No valid data	No valid data
Low	Low	High	Valid data bits 8-15	Valid data bits 0-7
High	Low	High	No valid data	Valid data bits 0-7
Low	High	High	Valid data bits 8-15	No valid data
Low	Low	Low	Valid data bits 8-15	Valid data bits 0-7
High	Low	Low	Valid data bits 0-7*	Valid data bits 0-7
Low	High	Low	Valid data bits 8-15	Valid data bits 8-15*

*These conditions are a result of current implementation and may not appear on future devices.

S5 and S6, and data will be captured during S6. The data must meet the required setup time (#27). If an asynchronous control signal does not meet the required setup time, it is possible that it may not be recognized during that cycle. Because of this, asynchronous systems must not allow DTACK to precede data by more than parameter #31.

Asserting DTACK (or BERR) on the rising edge of a clock (such as S4) after the assertion of address strobe will allow an SC68000 system to run at its maximum bus rate. If setup times #27 and #47 are guaranteed, #31 may be ignored.

Bus Arbitration Control

These three signals form a bus arbitration circuit to determine which device will be the bus master device:

Bus Request ($\overline{BR}$) — This input is wire ORed with all other devices that could be bus masters. It indicates to the processor that some other device desires to become the bus master.

Bus Grant ($\overline{BG}$) — This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

Bus Grant Acknowledge ($\overline{BGACK}$) — This input indicates that some other device has become the bus master. This signal cannot be asserted until the following four conditions are met:

1. A bus grant has been received.
2. Address strobe is inactive, indicating that the microprocessor is not using the bus.
3. Data transfer acknowledge is inactive, indicating that another device is not using the bus.
4. Bus grant acknowledge is inactive, indicating that no other device is still claiming bus mastership.

Interrupt Control ($\overline{IPL0}$,$\overline{IPL1}$,$\overline{IPL2}$)

These inputs indicate the encoded priority level of the device requesting an interrupt. Level seven is the highest priority while level zero indicates that no interrupts are requested. The least significant bit is given in IPL0 and the most signficant bit is contained in IPL2.

System Control

The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred.

Bus Error ($\overline{BERR}$) — This input informs the processor that there is a problem with the cycle currently being executed. Problems may be the result of nonresponding devices, interrupt vector acquisition failure, illegal access request as determined by a memory management unit, or other application dependent errors. The bus error signal interacts with the halt signal to determine if exception processing should be performed or the current bus cycle should be retried (see Bus Error and Halt Operation for additional information).

Reset ($\overline{RESET}$) — This bidirectional signal line acts to reset the processor (initiate a system initialization sequence) in response to an external reset signal. An internally generated reset (result of a RESET instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external HALT and RESET signals applied at the same time (see Reset Operation for additional information).

Halt ($\overline{HALT}$) — When this bidirectional line is driven by an external device, it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all three-state lines are put in their high-impedance state. When the processor has stopped executing instructions, such as in a double bus fault condition, the halt line is driven by the processor to indicate to external devices that the processor has stopped (see Bus Error and Halt Operation for additional information).

Peripheral Control

These control signals are used to allow the interfacing of synchronous peripheral devices with the asynchronous SC68000.

Enable (E) — This signal is the enable signal for synchronous type peripheral devices. The period for this output is ten SC68000 clock periods (six clocks low; four clocks high).

Valid Peripheral Address ($\overline{VPA}$) — This input indicates that the device or region addressed is a synchronous device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt (see Interface with Synchronous Peripherals for additional information).

Valid Memory Address ($\overline{VMA}$) — This output is used to indicate to synchronous peripheral devices that there is a valid address on the address bus and the processor is synchronized to enable. This signal is issued only in response to a valid peripheral address (VPA) input which indicates that the peripheral is a synchronous device.

Processor Status (FC0,FC1,FC2)

These function code outputs indicate the state (user or supervisor) and the cycle type currently being executed (see table 3). The information indicated by the function code is valid whenever address strobe (AS) is active.

Table 3. FUNCTION CODE OUTPUTS

FC2	FC1	FC0	Cycle Type
Low	Low	Low	(Undefined, Reserved)
Low	Low	High	User Data
Low	High	Low	User Program
Low	High	High	(Undefined, Reserved)
High	Low	Low	(Undefined, Reserved)
High	Low	High	Supervisor Data
High	High	Low	Supervisor Program
High	High	High	Interrupt Acknowledge

Clock (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input is a constant frequency.

REGISTER DESCRIPTION AND DATA ORGANIZATION

Operand Size

Operand sizes are defined as follows: a byte equals 8-bits,a word equals 16-bits, and a long word equals 32-bits. The operand size for each instruction is either explicitly encoded in the instruction or implicitly defined by the instruction operation. All explicit instructions support byte, word or long word operands. Implicit instructions support some subset of all three sizes.

Data Organization in Registers

The eight data registers support data operands of 1, 8, 16, or 32 bits. The seven address registers together with the active stack pointer support address operands of 32 bits.

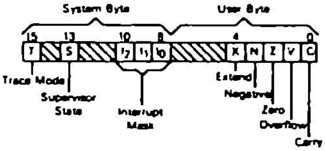
Data Registers — Each data register is 32-bits wide. Byte operands occupy the low order 8-bits, word operands the low order 16-bits, and long word operands the entire 32-bits. The least significant bit is addressed as bit 0, the most significant bit is addressed as bit 31. When a data register is used as either a source or destination operand and the operand size is not 32 bits, only the appropriate low-order portion is changed; the remaining high-order portion is neither used nor changed.

Address Registers — Each address register and the stack pointer are 32-bits wide and hold a full 32-bit address. Address registers do not support byte sized operands. Therefore, when an address register is used as a source operand, either the low order word or the entire long word operand is used depending on the operation size. When an address register is used as the destination operand, the entire register is affected regardless of the operation size. If the operation size is word, any other operands are sign extended to 32-bits before the operation is performed.

Status Register

The status register contains the interrupt mask (eight levels available) as well as the condition codes: extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate that the processor is in a trace (T) mode and/or in a supervisor (S) state.

STATUS REGISTER



Data Organization in Memory

Bytes are individually addressable with the high order byte having an even address the same as the word, as shown in figure 3. The low order byte has an odd address that is one count higher than the word address. Instructions and multibyte data are accessed only on word (even byte) boundaries. If a long word data is located at address n (n even), then the second word of that data is located at address n + 2.

The data types supported by the SC68000 are: bit data, integer data of 8, 16, or 32 bits, 32-bit addresses and binary coded decimal data. Each of these data types is put in memory as shown in figure 4.

BUS OPERATION

The following is an explanation of control signal and bus operation during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation.

Data Transfer Operations

Transfer of data between devices involves address bus A1-A23, data bus D0-D15 and control signals. The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for deskewing the acknowledge and data signals from the slave device.

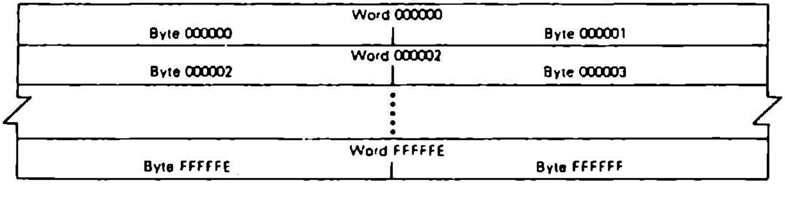


Figure 3. Word Organization in Memory

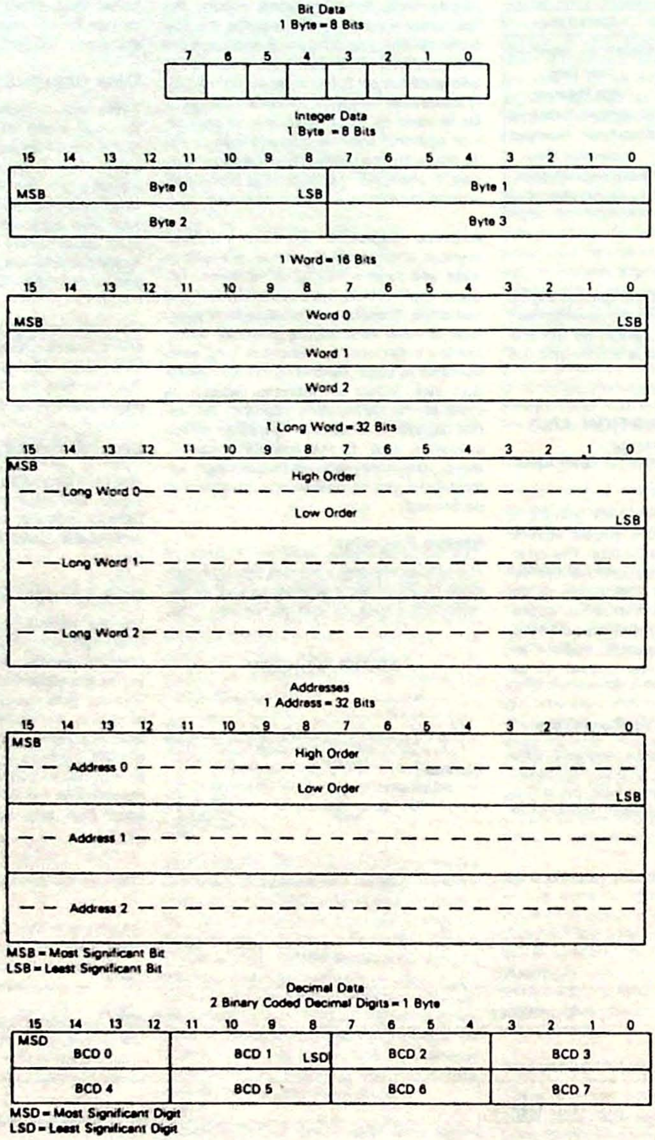

Bit Data
1 Byte = 8 Bits
7 6 5 4 3 2 1 0
Integer Data
1 Byte = 8 Bits
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
Byte 0
LSB
Byte 1
Byte 2
Byte 3
1 Word = 16 Bits
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
Word 0
LSB
Word 1
Word 2
1 Long Word = 32 Bits
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
High Order
Long Word 0
Low Order
LSB
Long Word 1
Long Word 2
Addresses
1 Address = 32 Bits
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
High Order
Address 0
Low Order
LSB
Address 1
Address 2
MSB = Most Significant Bit
LSB = Least Significant Bit
Decimal Data
2 Binary Coded Decimal Digits = 1 Byte
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSD
BCD 0
BCD 1
LSD
BCD 2
BCD 3
BCD 4
BCD 5
BCD 6
BCD 7
MSD = Most Significant Digit
LSD = Least Significant Digit

Figure 4. Data Organization In Memory

NOTE
The terms assertion and negation are used extensively to avoid confusion when dealing with a mixture of 'active low' and 'active high' signals. Assert or assertion is used to indicate that a signal is active or true independent of whether that voltage is low or high. Negate or negation is used to indicate that a signal is inactive or false.

Read Cycle — During a read cycle, the processor receives data from memory or a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or double word) operation, the processor reads both bytes simultaneously. When the instruction specifies byte operation, the processor uses an internal A0 bit to determine which byte to read and then issues the data strobe required for that byte: when the A0 bit equals zero, the upper data strobe is issued; when the A0 bit equals one, the lower data strobe is issued. When the data is received, the processor correctly positions it internally.

Flow charts for word and byte read cycles are shown in figures 5 and 6 respectively. Figure 7 illustrates read and write cycle timing and figure 8 illustrates the timing for word and byte read cycles.

Write Cycle — During a write cycle, the processor sends data to memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes simultaneously. When the instruction specifies a byte operation, the processor uses an internal A0 bit to determine which byte to write and then issues the data strobe required for that byte: when the A0 bit equals zero, the upper data strobe is issued; when the A0 bit equals one, the lower data strobe is issued.

Flow charts for word and byte write cycles are shown in figures 9 and 10 respectively. Figure 11 illustrates the timing for both cases.

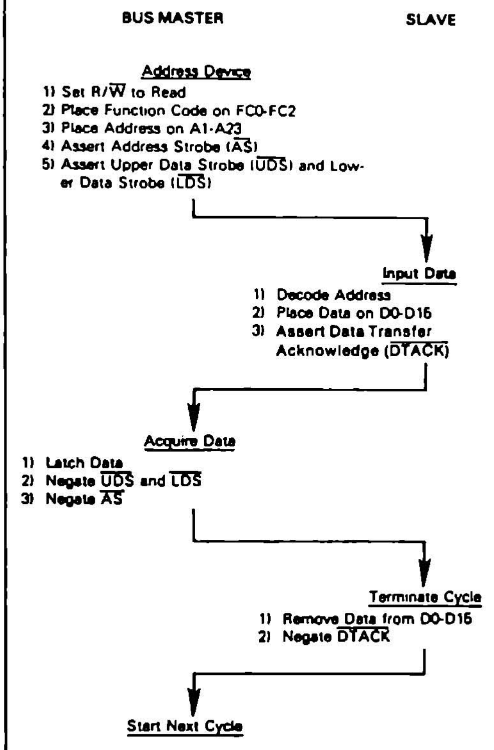

BUS MASTER
SLAVE
Address Device
1) Set R/$\overline{W}$ to Read
2) Place Function Code on FC0-FC2
3) Place Address on A1-A23
4) Assert Address Strobe ($\overline{AS}$)
5) Assert Upper Data Strobe ($\overline{UDS}$) and Lower Data Strobe ($\overline{LDS}$)
Input Data
1) Decode Address
2) Place Data on D0-D15
3) Assert Data Transfer Acknowledge ($\overline{DTACK}$)
Acquire Data
1) Latch Data
2) Negate $\overline{UDS}$ and $\overline{LDS}$
3) Negate $\overline{AS}$
Terminate Cycle
1) Remove Data from D0-D15
2) Negate $\overline{DTACK}$
Start Next Cycle

Figure 5. Word Read Cycle Flow Chart

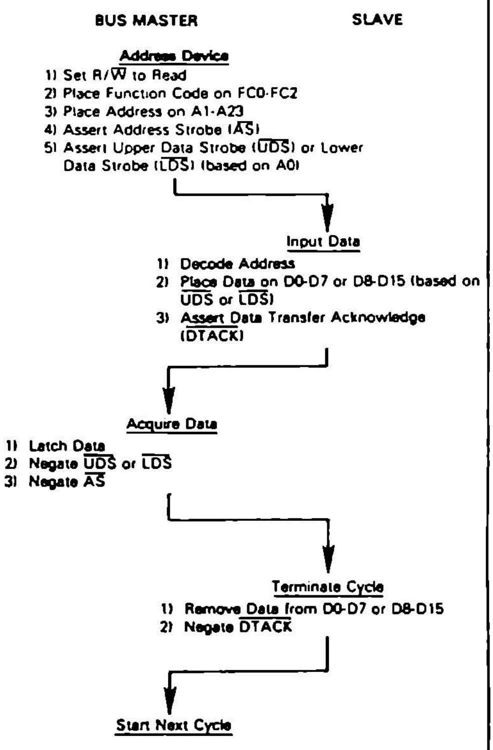

BUS MASTER
SLAVE
Address Device
1) Set R/$\overline{W}$ to Read
2) Place Function Code on FC0-FC2
3) Place Address on A1-A23
4) Assert Address Strobe ($\overline{AS}$)
5) Assert Upper Data Strobe ($\overline{UDS}$) or Lower Data Strobe ($\overline{LDS}$) (based on A0)
Input Data
1) Decode Address
2) Place Data on D0-D7 or D8-D15 (based on $\overline{UDS}$ or $\overline{LDS}$)
3) Assert Data Transfer Acknowledge ($\overline{DTACK}$)
Acquire Data
1) Latch Data
2) Negate $\overline{UDS}$ or $\overline{LDS}$
3) Negate $\overline{AS}$
Terminate Cycle
1) Remove Data from D0-D7 or D8-D15
2) Negate $\overline{DTACK}$
Start Next Cycle

Figure 6. Byte Read Cycle Flow Chart

Preliminary

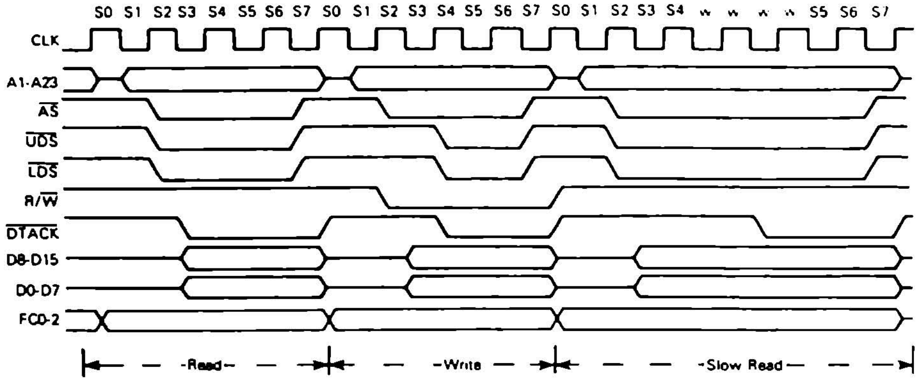
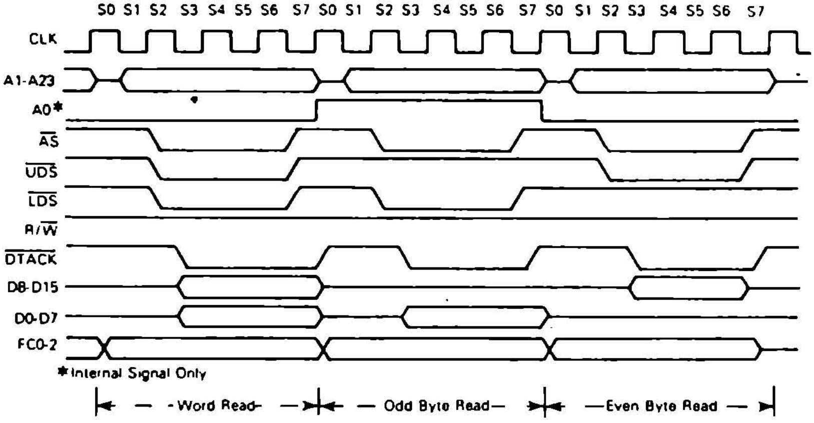
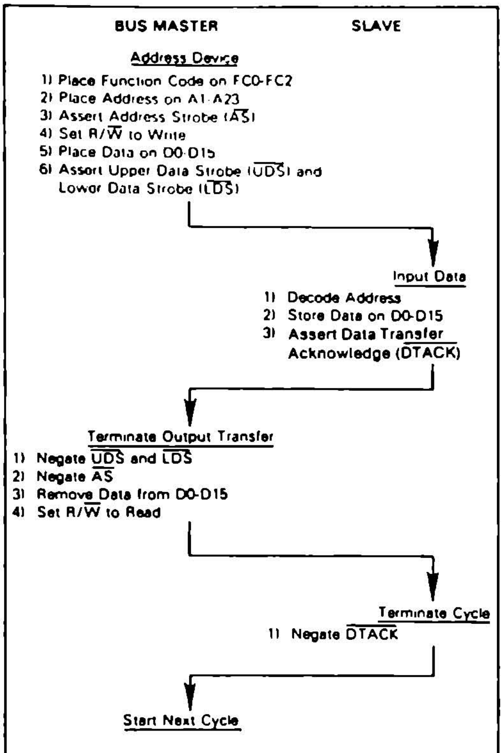

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 w w w w S5 S6 S7
CLK
A1-A23
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
FC0-2
Read
Write
Slow Read

Figure 7. Read and Write Cycle Timing


S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
CLK
A1-A23
A0*
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
FC0-2
Word Read
Odd Byte Read
Even Byte Read

*Internal Signal Only

Figure 8. Word and Byte Read Cycle Timing



BUS MASTER — **SLAVE**

Address Device

1) Place Function Code on FC0-FC2
2) Place Address on A1-A23
3) Assert Address Strobe ($\overline{AS}$)
4) Set R/$\overline{W}$ to Write
5) Place Data on D0-D15
6) Assert Upper Data Strobe ($\overline{UDS}$) and Lower Data Strobe ($\overline{LDS}$)

Input Data

1) Decode Address
2) Store Data on D0-D15
3) Assert Data Transfer Acknowledge ($\overline{DTACK}$)

Terminate Output Transfer

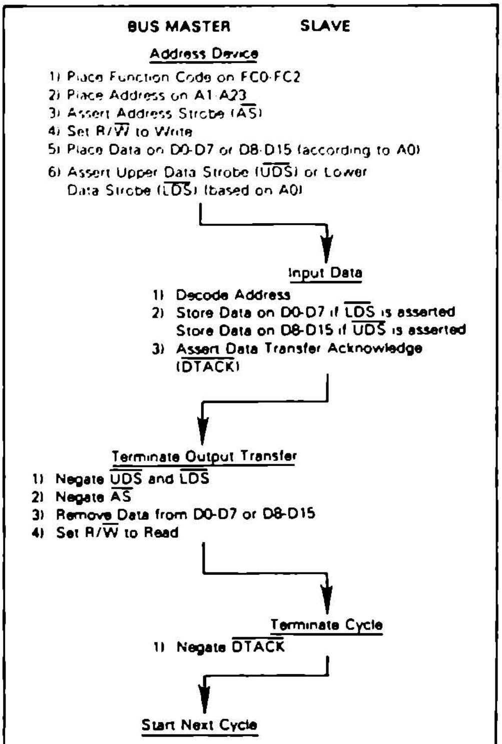
1) Negate $\overline{UDS}$ and $\overline{LDS}$
2) Negate $\overline{AS}$
3) Remove Data from D0-D15
4) Set R/$\overline{W}$ to Read

Terminate Cycle

1) Negate $\overline{DTACK}$

Start Next Cycle

Figure 9. Word Write Cycle Flow Chart



BUS MASTER — **SLAVE**

Address Device

1) Place Function Code on FC0-FC2
2) Place Address on A1-A23
3) Assert Address Strobe ($\overline{AS}$)
4) Set R/$\overline{W}$ to Write
5) Place Data on D0-D7 or D8-D15 (according to A0)
6) Assert Upper Data Strobe ($\overline{UDS}$) or Lower Data Strobe ($\overline{LDS}$) (based on A0)

Input Data

1) Decode Address
2) Store Data on D0-D7 if $\overline{LDS}$ is asserted
   Store Data on D8-D15 if $\overline{UDS}$ is asserted
3) Assert Data Transfer Acknowledge ($\overline{DTACK}$)

Terminate Output Transfer

1) Negate $\overline{UDS}$ and $\overline{LDS}$
2) Negate $\overline{AS}$
3) Remove Data from D0-D7 or D8-D15
4) Set R/$\overline{W}$ to Read

Terminate Cycle

1) Negate $\overline{DTACK}$

Start Next Cycle

Figure 10. Byte Write Cycle Flow Chart

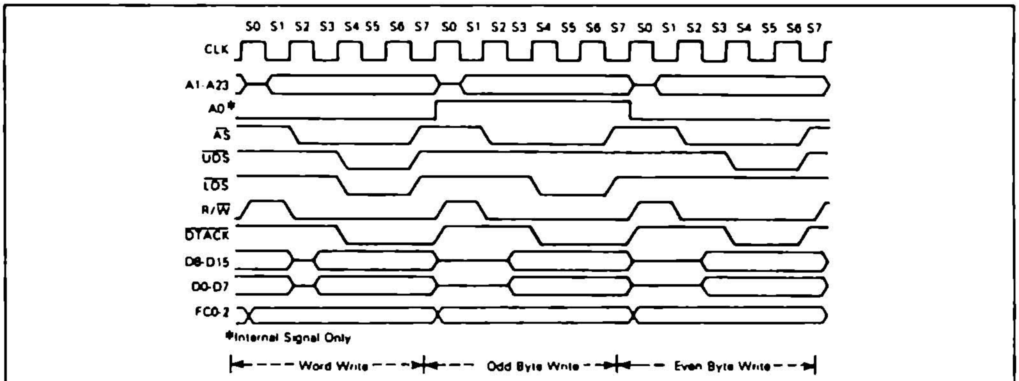

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
CLK
A1-A23
A0*
$\overline{AS}$
$\overline{UDS}$
$\overline{LDS}$
R/$\overline{W}$
$\overline{DTACK}$
D8-D15
D0-D7
FC0-2
*Internal Signal Only
Word Write
Odd Byte Write
Even Byte Write

Figure 11. Word and Byte Write Cycle Timing

Read-Modify-Write Cycle — The read-modify-write cycle performs a read, modifies the data in the arithmetic-logic unit, and writes the data back to the same address. In the SC68000, this cycle is indivisible in that the address strobe is asserted throughout the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment, and is the only instruction that uses it. The read-modify-write cycle is always a byte operation. The flow chart is given in figure 12 and a timing diagram is shown in figure 13.

Bus Arbitration

Bus arbitration is a technique used by master type devices to request, be granted, and acknowledge bus mastership. In its simplist form, it consists of:

1. Asserting a bus mastership request.
2. Receiving a grant that the bus is available at the end of the current cycle.
3. Acknowledging that mastership has been assumed.

Figure 14 is a flow chart showing the detail involved in a request from a single device. Figure 15 is a timing diagram for the same operations. The technique used allows processing of bus requests during data transfer cycles.

The timing diagram shows that the bus request is negated at the time that an acknowledge is asserted. This type of operation would be true for a system consisting of the processor and one device capable of bus mastership. In systems having a number of devices capable of mastership, the bus request line from each device is wire-ORed to the processor. In this system, there could be more than

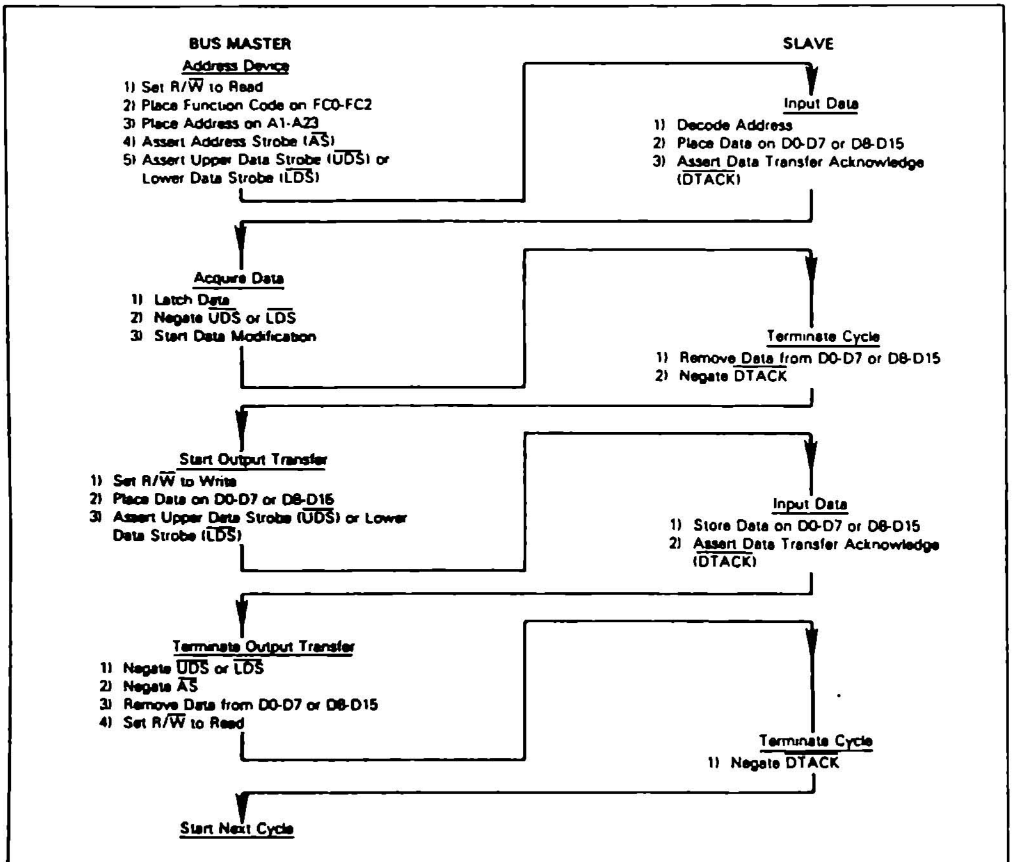

BUS MASTER
SLAVE
Address Device
1) Set R/W to Read
2) Place Function Code on FC0-FC2
3) Place Address on A1-A23
4) Assert Address Strobe (AS)
5) Assert Upper Data Strobe (UDS) or Lower Data Strobe (LDS)
Input Data
1) Decode Address
2) Place Data on D0-D7 or D8-D15
3) Assert Data Transfer Acknowledge (DTACK)
Acquire Data
1) Latch Data
2) Negate UDS or LDS
3) Start Data Modification
Terminate Cycle
1) Remove Data from D0-D7 or D8-D15
2) Negate DTACK
Start Output Transfer
1) Set R/W to Write
2) Place Data on D0-D7 or D8-D15
3) Assert Upper Data Strobe (UDS) or Lower Data Strobe (LDS)
Input Data
1) Store Data on D0-D7 or D8-D15
2) Assert Data Transfer Acknowledge (DTACK)
Terminate Output Transfer
1) Negate UDS or LDS
2) Negate AS
3) Remove Data from D0-D7 or D8-D15
4) Set R/W to Read
Terminate Cycle
1) Negate DTACK
Start Next Cycle

Figure 12. Read-Modify-Write Cycle Flow Chart

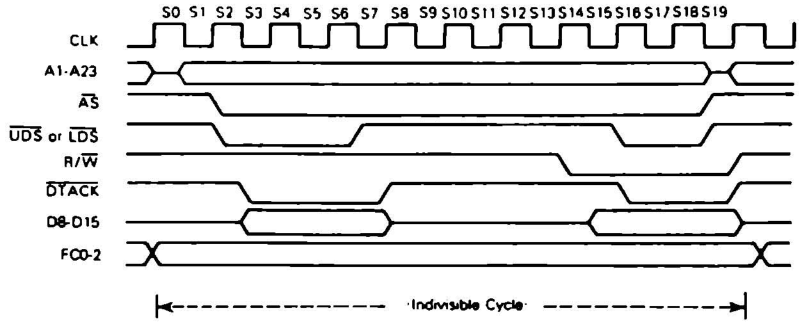

S0 S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19
CLK
A1-A23
AS
UDS or LDS
R/W
DTACK
D8-D15
FC0-2
Indivisible Cycle

Figure 13. Read-Modify-Write Cycle Timing

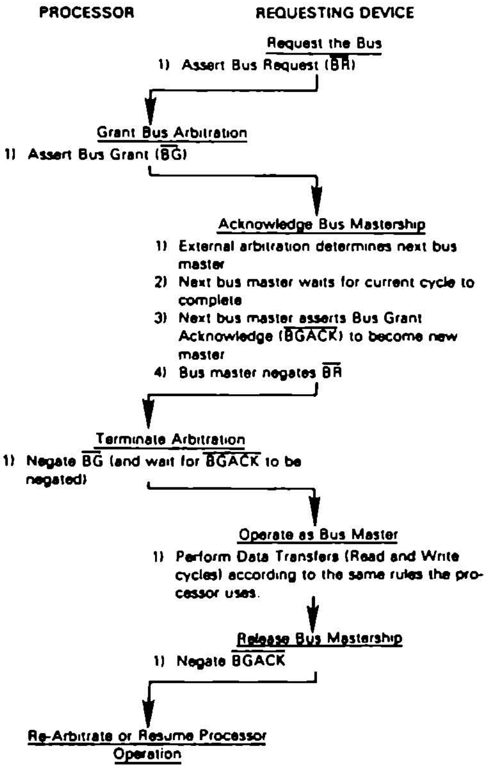

PROCESSOR
REQUESTING DEVICE
Request the Bus
1) Assert Bus Request (BR)
Grant Bus Arbitration
1) Assert Bus Grant (BG)
Acknowledge Bus Mastership
1) External arbitration determines next bus master
2) Next bus master waits for current cycle to complete
3) Next bus master asserts Bus Grant Acknowledge (BGACK) to become new master
4) Bus master negates BR
Terminate Arbitration
1) Negate BG (and wait for BGACK to be negated)
Operate as Bus Master
1) Perform Data Transfers (Read and Write cycles) according to the same rules the processor uses.
Release Bus Mastership
1) Negate BGACK
Re-Arbitrate or Resume Processor Operation

Figure 14. Bus Arbitration Cycle Flow Chart

one bus request being made. The timing diagram shows that the bus grant signal is negated a few clock cycles after the transition of the acknowledge (BGACK) signal. However, if the bus requests are still pending, the processor will assert another bus grant within a few clock cycles after it was negated. This additional assertion of bus grant allows external arbitration circuitry to select the next bus master before the current bus master has completed its requirements.

Requesting the Bus — External devices, capable of becoming bus masters, request the bus by asserting the bus request (BR) signal. This is a wire-ORed signal (although it need not be constructed from open collector devices) that indicates to the processor that some external device requires control of the external bus. The processor is effectively at a lower bus priority level than the external device and will relinquish the bus after it has completed the last bus cycle it has started.

When no acknowledge is received before the bus request signal goes inactive, the processor will continue processing when it detects that the bus request is inactive. This allows ordinary processing to continue if the arbitration circuitry inadvertently responded to noise.

Receiving the Bus Grant — The processor asserts bus grant (BG) as soon as possible. Normally this is immediately after internal synchronization. The only exception to this occurs when the processor has made an internal decision to execute the next bus cycle but has not progressed far enough into the cycle to have asserted the address strobe (AS) signal. In this case, bus grant will not be asserted until one

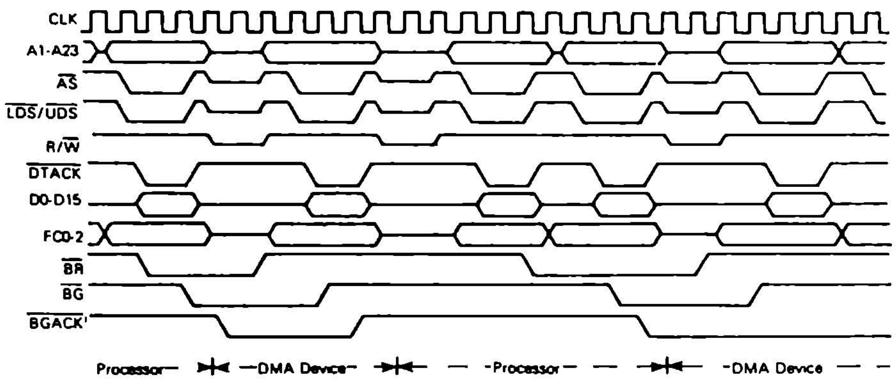


Figure 15. Bus Arbitration Cycle Timing

clock after address strobe is asserted to indicate to external devices that a bus cycle is being executed.

The bus grant signal may be routed through a daisy-chained network or through a specific priority-encoded network. The processor is not affected by the external method of arbitration as long as the protocol is obeyed.

Acknowledgement of Mastership — Upon receiving a bus grant, the requesting device waits until address strobe, data transfer acknowledge, and bus grant acknowledge are negated before issuing its own BGACK. The negation of the address strobe indicates that the previous master has completed its cycle, and the negation of bus grant acknowledge indicates that the previous master has released the bus. A device is not allowed to 'break into' a cycle while address strobe is asserted. The negation of data transfer acknowledge indicates that the previous slave has terminated its connection to the previous master. Note that in some applications, data transfer acknowledge might not enter into this function. General purpose devices would then be connected such that they were only dependent on address strobe. When bus grant acknowledge is issued, the device remains the bus master until it negates bus grant acknowledge. Bus grant acknowledge should not be negated until after the bus cycle(s) is completed.

The bus request from the granted device should be dropped when bus grant acknowledge is asserted. If bus request is still asserted after bus grant acknowledge is negated, the processor performs another arbitration sequence and issues another bus grant. Note that the processor does not perform any external bus cycles before it reasserts bus grant.

Bus Arbitration Control — The bus arbitration control unit in the SC68000 is implemented with a finite state machine whose state diagram is shown in figure 16. All asynchronous signals to the SC68000 are synchronized before being used internally. This synchronization is accomplished in a maximum of one cycle of the system clock, assuming that the asynchronous input setup time (#47) has been met (see figure 17). The input signal is sampled on the falling edge of the clock and is valid internally after the next falling edge.

As shown in figure 16, input signals labeled R and A are internally synchronized on the bus request and bus grant acknowledge pins respectively. The bus grant output is labeled G and the internal three-state control signal T. If T is true, the address, data, and control buses are placed in a high-impedance state when AS is negated. All signals are shown in positive logic (active high) regardless of their true active voltage level.

State changes (valid outputs) occur on the next rising edge after the internal signal is valid.

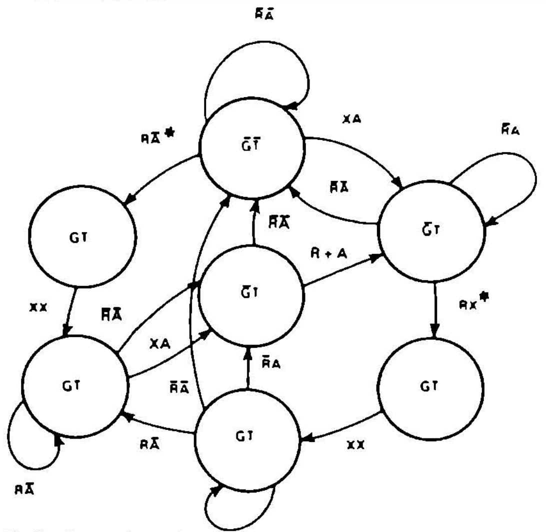
A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in figure 18. The bus arbitration sequence while the bus is inactive (i.e., executing internal operations such as a multiply instruction) is shown in figure 19.

If a bus request is made at a time when the MPU has already begun a bus cycle but AS has not been asserted (bus state S0), BG will not be asserted on the next rising edge following its internal assertion. This sequence is shown in figure 20.

Bus Error and Halt Operation

In a bus architecture that requires a handshake from an external device, the possibility exists that the handshake might not occur. Since different systems will require a different maximum response time, a bus error input is provided. External circuitry must be used to determine the duration between address strobe and data transfer acknowledge before issuing a bus error signal. When a bus error signal is received, the processor has the option of either initiating a bus error exception sequence or trying to run the bus cycle again.

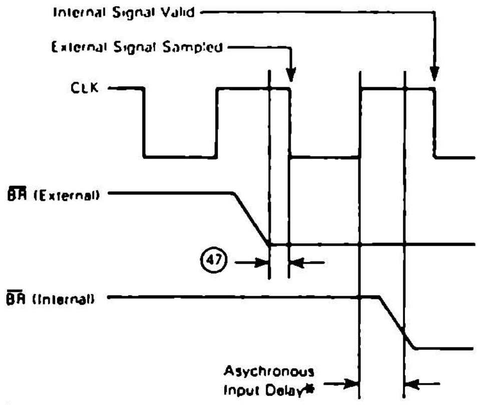
Exception Sequence — When the bus error signal is asserted, the current bus cycle is terminated. If BERR is asserted before the falling edge of S4, AS will be negated in S7 in either a read or write cycle. As long as BERR remains asserted, the data and address buses will be in the high-impedance state. When BERR is negated, the processor will begin stacking for the exception sequence. Figure 21 is a timing diagram for the exception se-



R = Bus Request Internal
A = Bus Grant Acknowledge Internal
G = Bus Grant
T = Three-State Control to Bus Control Logic
X = Don't Care

* State machine will not change state if bus is in S0. Refer to BUS ARBITRATION CONTROL for additional information.

Figure 16. State Diagram of SC68000 Bus Arbitration Unit



*This delay time is equal to parameter #33, tCHGL

Figure 17. Timing Relationship of External Inputs to Internal Signals

quence which is composed of the following elements:

1. Stacking the program counter and status register.
2. Stacking the error information.
3. Reading the bus error vector table entry.
4. Executing the bus error handling routine.

The stacking of the program counter and the status register is the same as if an interrupt had occurred. Several additional items are stacked when a bus error occurs to determine the nature of the error and to correct it, if possible. The bus error vector is vector number two located at address $000008. The processor loads the program counter from this location and a software bus error handler routine is then executed by the processor. Refer to Exception Processing for additional information.

Rerunning the Bus Cycle — When the processor receives a bus error signal and the halt pin is being driven by an external device, the processor enters the rerun sequence illustrated in figure 22.

The processor completes the bus cycle, then puts the address, data, function code, and control leads in the high-impedance state. The processor remains 'halted' and will not run another bus cycle until the halt signal is removed by external logic. The processor will then rerun the previous bus cycle using the same address, the same function codes, the same data (for a write operation), and the same controls. The bus error signal should be removed before the halt signal is removed.

NOTE
The processor will not rerun a read-modify-write cycle. This restriction is made to guarantee that the entire cycle runs correctly and that the write operation of a test-and-set operation is performed without ever releasing AS. If BERR and HALT are asserted during a read-modify-write cycle, a bus error operation results.

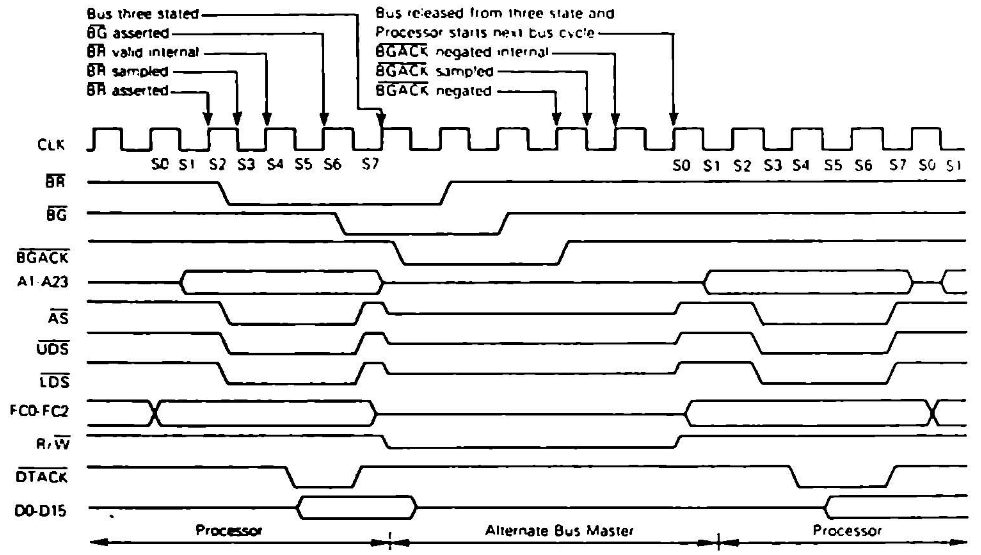

Bus three stated
BG asserted
BR valid internal
BR sampled
BR asserted
Bus released from three state and
Processor starts next bus cycle
BGACK negated internal
BGACK sampled
BGACK negated
CLK
S0 S1 S2 S3 S4 S5 S6 S7
S0 S1 S2 S3 S4 S5 S6 S7 S0 S1
BR
BG
BGACK
A1-A23
AS
UDS
LDS
FC0-FC2
R/W
DTACK
D0-D15
Processor
Alternate Bus Master
Processor

Figure 18. Bus Arbitration During Processor Bus Cycle

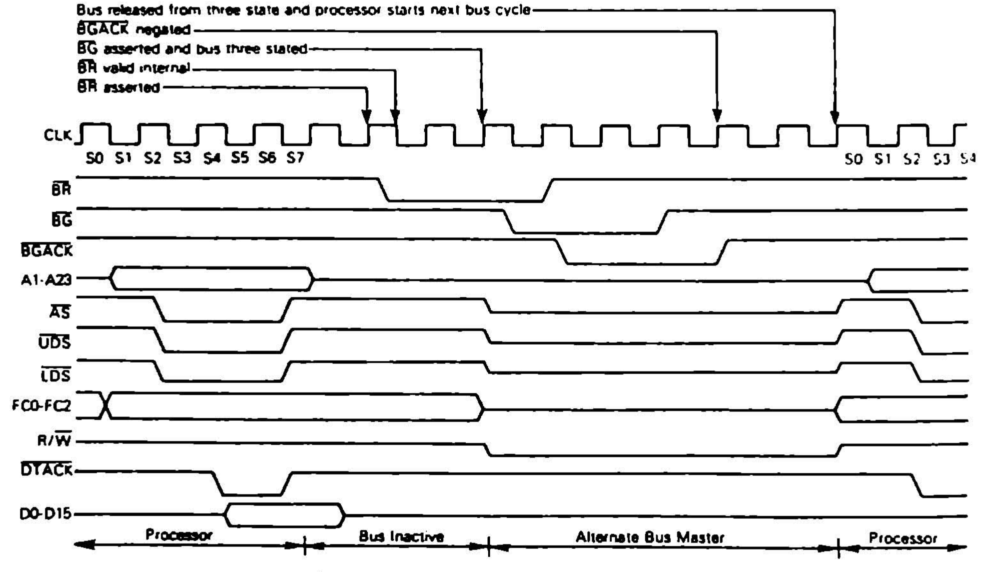

Bus released from three state and processor starts next bus cycle
BGACK negated
BG asserted and bus three stated
BR valid internal
BR asserted
CLK
S0 S1 S2 S3 S4 S5 S6 S7
S0 S1 S2 S3 S4
BR
BG
BGACK
A1-A23
AS
UDS
LDS
FC0-FC2
R/W
DTACK
D0-D15
Processor
Bus Inactive
Alternate Bus Master
Processor

Figure 19. Bus Arbitration with Bus Inactive

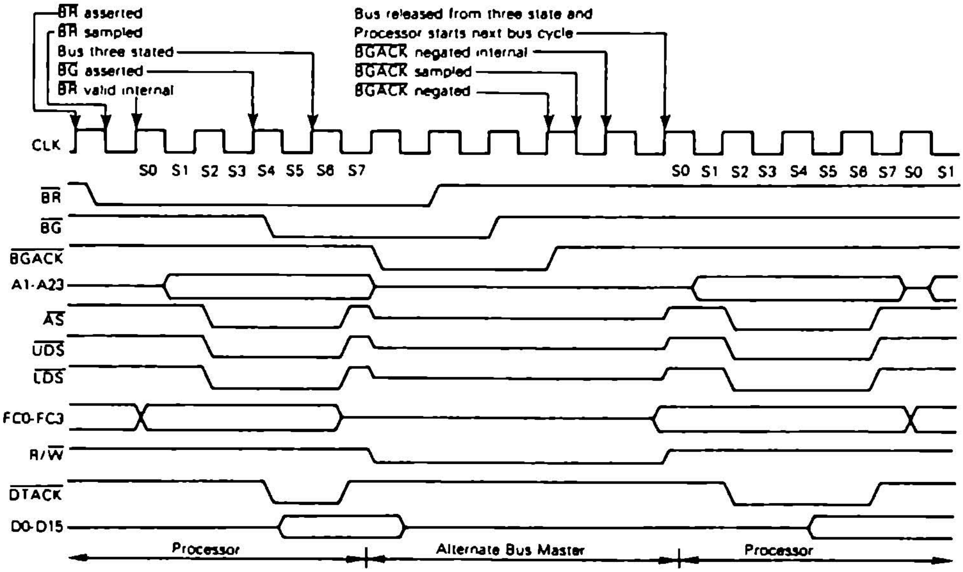

$\overline{BR}$ asserted
$\overline{BR}$ sampled
Bus three stated
$\overline{BG}$ asserted
$\overline{BR}$ valid internal
Bus released from three state and Processor starts next bus cycle
$\overline{BGACK}$ negated internal
$\overline{BGACK}$ sampled
$\overline{BGACK}$ negated
CLK
S0 S1 S2 S3 S4 S5 S6 S7
S0 S1 S2 S3 S4 S5 S6 S7 S0 S1
$\overline{BR}$
$\overline{BG}$
$\overline{BGACK}$
A1-A23
$\overline{AS}$
$\overline{UDS}$
$\overline{LDS}$
FC0-FC3
R/$\overline{W}$
$\overline{DTACK}$
D0-D15
Processor
Alternate Bus Master
Processor

Figure 20. Bus Arbitration During Processor Bus Cycle Special Case

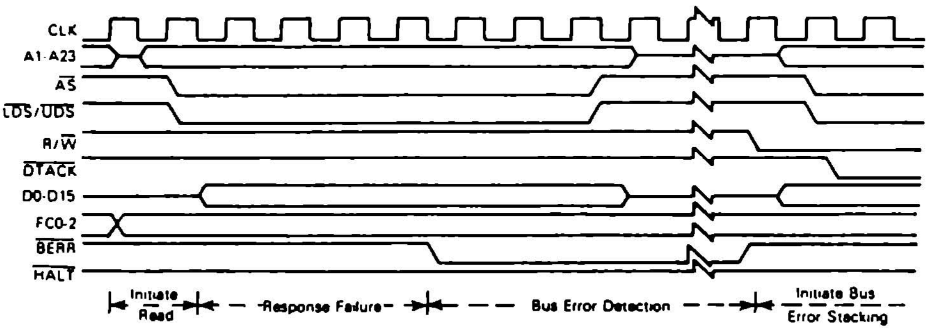

CLK
A1-A23
$\overline{AS}$
$\overline{LDS}$/$\overline{UDS}$
R/$\overline{W}$
$\overline{DTACK}$
D0-D15
FC0-2
$\overline{BERR}$
$\overline{HALT}$
Initiate Read
Response Failure
Bus Error Detection
Initiate Bus Error Stacking

Figure 21. Bus Error Timing

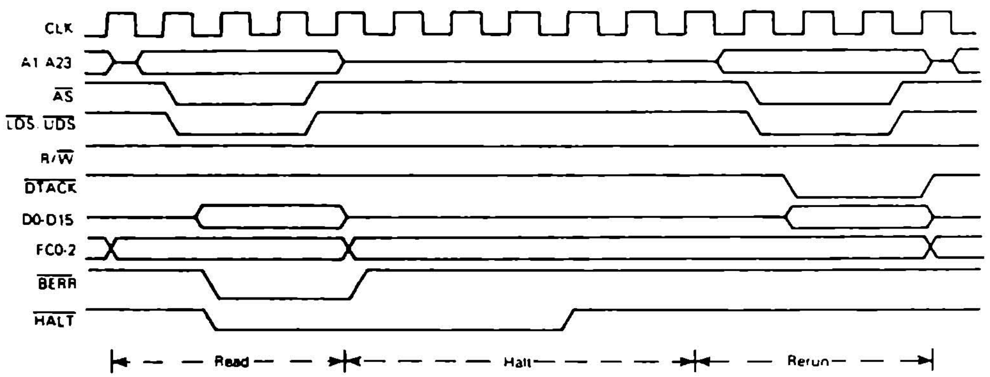


Figure 22. Rerun Bus Cycle Timing

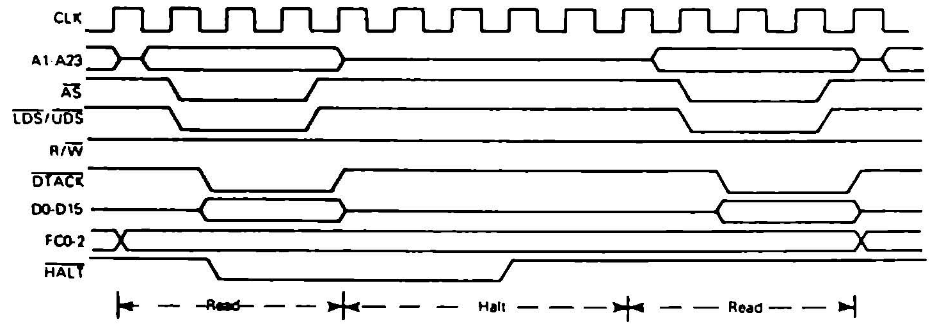


Figure 23. Halt Signal Timing

Halt Operation with No Bus Error — The halt input signal to the SC68000 can be used to perform a halt/run/single-step function. The halt and run modes are somewhat self explanatory in that, when the halt signal is constantly active, the processor 'halts' (does nothing) and when the halt signal is constantly inactive, the processor 'runs' (does something).

The single-step mode is derived from correctly timed transitions on the halt signal input. It forces the processor to execute a single bus cycle by entering the 'run' mode until the processor starts a bus cycle, then changing to the 'halt' mode. Thus, the single step mode allows the user to proceed through (and therefore debug) processor operations one bus cycle at a time.

Figure 23 shows the timing required for correct single-step operations. Some care must be exercised to avoid harmful interactions between the bus error signal and the halt pin when using the single cycle mode as a debugging tool. This is also true of interactions between the halt and reset lines since these can reset the processor.

When the processor completes a bus cycle after recognizing that the halt signal is active, most three-state signals are put in the high-impedance state. These include address lines and data lines. This is required for correct performance of the rerun bus cycle operation.

While the processor is honoring the halt request, bus arbitration performs as usual. That is, halting has no effect on bus arbitration. It is the bus arbitration function that removes the control signals from the bus.

The halt function and the hardware trace capability allow the hardware debugger to trace single bus cycles or single instructions at a time. These processor capabilities, along with a software debugging package, give total debugging flexibility.

Double Bus Faults — When a bus error exception occurs, the processor will attempt to stack several words containing information about the state of the processor. If a bus error exception occurs during the stacking operation, there have been two bus errors in a row, which is commonly referred to as a double bus fault. When a double bus fault occurs, the processor will halt. Once a bus error exception has occurred, any bus error exception occurring before the execution of the next instruction constitutes a double bus fault.

Note that a bus cycle which is rerun does not constitute a bus error exception, and does not contribute to a double bus fault. This means that as long as the external hardware requests it, the processor will continue to rerun the same bus cycle.

The bus error pin also has an effect on the processor operation after the processor receives an external reset input. The processor reads the vector table after a reset to determine the address to start program execution. If a bus error occurs while reading the vector table (or at any time before the first instruction is executed), the processor reacts as if a double bus fault has occurred and it halts. Only an external reset will start a halted processor.

Relationship of DTACK, BERR, and HALT

In order to properly control termination of a bus cycle for a rerun or a bus error condition, DTACK, BERR, and HALT should be asserted and negated on the rising edge of the SC68000 clock. This will assure that when two signals are asserted simultaneously, the required setup time (#47) for both of them will be met during the same bus state.

This, or some equivalent precaution, should be designed external to the SC68000. Parameter #48 is intended to ensure this operation in a totally asynchronous system, and may be ignored if the above conditions are met.

The preferred bus cycle terminations can be summarized as follows (case numbers refer to table 4):

Normal termination: DTACK occurs first (case 1).

Halt termination: HALT is asserted at the same time, or precedes DTACK (no BERR) cases 2 and 3.

Bus error termination: BERR is asserted in lieu of, at same time, or preceding DTACK (case 4); BERR negated at same time, or after DTACK.

Rerun termination: HALT and BERR asserted at the same time, or before DTACK (cases 6 and 7); HALT must be negated at least one cycle after BERR. Case 5 indicates BERR can precede HALT which allows fully asynchronous assertion.

Table 4 details the resulting bus cycle termination under various combinations of control signal sequences. The negation of these same control signals under several conditions is shown in table 5 (DTACK is assumed to be negated normally in all cases; for best results, both DTACK and BERR should be negated when address strobe is negated).

Example A: A system uses a watch-dog timer to terminate accesses to unpopulated address space. The timer asserts DTACK and BERR simultaneousl after time-out (case 4).

Table 4. DTACK, BERR, HALT ASSERTION RESULTS

Case No.	Control Signal	Asserted on Rising Edge of State		Result
		N	N + 2	
1	$\overline{\text{DTACK}}$	A	S	Normal cycle terminate and continue
	$\overline{\text{BERR}}$	NA	X	
	$\overline{\text{HALT}}$	NA	X	
2	$\overline{\text{DTACK}}$	A	S	Normal cycle terminate and halt Continue when HALT removed.
	$\overline{\text{BERR}}$	NA	X	
	$\overline{\text{HALT}}$	A	S	
3	$\overline{\text{DTACK}}$	NA	A	Normal cycle terminate and halt. Continue when HALT removed.
	$\overline{\text{BERR}}$	NA	NA	
	$\overline{\text{HALT}}$	A	S	
4	$\overline{\text{DTACK}}$	X	X	Terminate and take bus error trap.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	NA	NA	
5	$\overline{\text{DTACK}}$	NA	X	Terminate and re-run.
	$\overline{\text{BERR}}$	A	S	
	HALT	NA	A	
6	$\overline{\text{DTACK}}$	X	X	Terminate and re-run.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	A	S	
7	$\overline{\text{DTACK}}$	NA	X	Terminate and re-run when HALT removed
	$\overline{\text{BERR}}$	NA	A	
	$\overline{\text{HALT}}$	A	S	

Legend:
N – the number of the current even bus state (e.g., S4, S6, etc.)
A – signal is asserted in this bus state
NA – signal is not asserted in this state
X – don't care
S – signal was asserted in previous state and remains asserted in this state

Example 8: A system uses error detection on RAM contents. Designer can (a) delay DTACK until the data is verified, and return BERR and HALT simultaneously to rerun error cycle (case 6), or if valid, return DTACK; (b) delay DTACK until data is verified, and return BERR at the same time as DTACK if data is in error (case 4); (c) return DTACK prior to data verification, as described in the previous section. If data is invalid, BERR is asserted (case 1) in the next cycle. Error handling software must know how to recover the error cycle.

Reset Operation

The reset signal is a bidirectional signal that allows either the processor or an external signal to reset the system. Figure 24 illustrates reset timing. Both the halt and the reset lines must be applied to ensure total reset of the processor.

When the reset and halt lines are driven by an external device, it is recognized as an entire system reset, including the processor. The processor responds by reading the reset vector table entry (vector number zero, address $000000) and loads it into the supervisor stack pointer (SSP). Vector table entry number one at address $000004 is read next and loaded into the program counter. The processor initializes the status register to an interrupt level of seven. No other registers are affected by the reset sequence.

When a RESET instruction is executed, the processor drives the reset pin for 124 clock pulses. In this case, the processor is trying to reset the rest of the system. Therefore, there is no effect on the internal state of the processor and its internal registers and the status register are unaffected. All external devices connected to the reset line should be reset at the completion of the RESET instruction.

Asserting the RESET and HALT pins for ten clock cycles will cause a processor reset, except when V_{CC} is initially applied to the processor. In this case, an external reset must be applied for 100 milliseconds.

PROCESSING STATES

The SC68000 is always in one of three processing states: normal, exception, or halted. The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. A special case of the normal state is the stopped state which the processor enters when a STOP instruction is executed. In this state, no further memory references are made.

The exception processing state is associated with interrupts, trap instructions, tracing and other exceptional conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is designed to provide an efficient context switch so that the processor can handle unusual conditions.

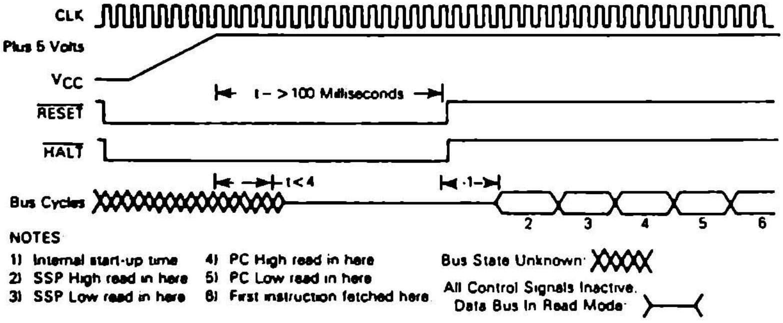
The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

Privilege States

The processor operates in one of two states of privilege: the 'user' state or the 'supervisor' state. The privilege state determines which operations are legal, is used by the external memory management device to control and translate accesses.

Table 5. BERR AND HALT NEGATION RESULTS

Conditions of Termination in Table A	Control Signal	Negated on Rising Edge of State			Results — Next Cycle
		N		N + 2	
Bus Error	$\overline{\text{BERR}}$	●	or	●	Takes bus error trap
	$\overline{\text{HALT}}$	●	or	●	
Re-run	$\overline{\text{BERR}}$	●	or	●	Illegal sequence, usually traps to vector number 0
	$\overline{\text{HALT}}$	●			
Re-run	$\overline{\text{BERR}}$	●			Re-runs the bus cycle
	$\overline{\text{HALT}}$			●	
Normal	$\overline{\text{BERR}}$	●			May lengthen next cycle
	$\overline{\text{HALT}}$	●	or	●	
Normal	$\overline{\text{BERR}}$			●	If next cycle is started it will be terminated as a bus error
	$\overline{\text{HALT}}$	●	or	none	



NOTES
1) Internal start-up time
2) SSP High read in here
3) SSP Low read in here
4) PC High read in here
5) PC Low read in here
6) First instruction fetched here

Figure 24. Reset Operation Timing

and is used to choose between the supervisor stack pointer and the user stack pointer in instruction references.

The privilege state is a mechanism for providing security in a computer system. Programs should access only their own code and data areas, and ought to be restricted from accessing information which they do not need and must not modify.

The privilege mechanism provides security by allowing most programs to execute in user state. In this state, their accesses are controlled, and the effects on other parts of the system are limited. The operating system executes in the supervisor state, has access to all resources, and performs the overhead tasks for the user state programs.

Supervisor State — The supervisor state is the higher state of privilege. For instruction execution, the supervisor state is determined by the S-bit of the status register: it is asserted (high), the processor is in the supervisor state. All instructions can be executed in the supervisor state. The bus cycles generated by instructions executed in the supervisor state are classified as supervisor references. While the processor is in the supervisor privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly access the supervisor stack pointer.

All exception processing is done in the supervisor state, regardless of the setting of the S-bit. The bus cycles generated during exception processing are classified as supervisor references. All stacking operations during exception processing use the supervisor stack pointer.

User State — The user state is the lower state of privilege. For instruction execution, the user state is determined by the S-bit of the status register: it is negated (low), the processor is executing instructions in the user state.

Most instructions execute the same in user state as in the supervisor state. However, some instructions which have important system effects are made privileged. User programs are not permitted to execute the STOP instruction, or the RESET instruction. To ensure that a user program cannot enter the supervisor state except in a controlled manner, the instructions which modify the whole status register are privileged. To aid in debugging programs which are to be used as operating systems, the move to user stack pointer (MOVE to USP) and move from user stack pointer (MOVE and USP) instructions are also privileged.

The bus cycles generated by an instruction executed in user state are classified as user state references. This allows an external memory management device to translate the address and to control access to protected portions of the address space. While the processor is in the user privilege state, those instructions which use either the system stack pointer implicitly, or address register seven explicitly, access the user stack pointer.

Privilege State Changes — Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the current setting of the S-bit of the status register is saved and the S-bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state.

Reference Classification — When the processor makes a reference, it classifies the kind of reference being made, using the encoding on the three function code output lines. This allows external translation of addresses, control of access, and differentiation of special processor states, such as an interrupt acknowledge. Table 6 lists the classification of references.

Exception Processing General Description

The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the status register is made, and the status register is set for exception processing. In the second step the exception vector is determined, and the third step is the saving of the current processor context. In the fourth step a new context is obtained, and the processor switches to instruction processing.

Exception Vectors — Exception vectors are memory locations from which the processor fetches the address of a routine which will handle that exception. All exception vectors are two words in length (see figure 25), except for the reset vector, which is four words. All exception vectors lie in the supervisor data space, except for the reset vector which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the address of an exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of interrupts, during the interrupt acknowledge bus cycle, a peripheral provides an 8-bit vector number (see figure 26) to the processor on the data bus lines D0 through D7. The processor translates the vector number into a full 24-bit address, as shown in figure 27. The memory layout for exception vectors is given in table 7.

As shown in table 7, the memory layout is 512 words long (1024 bytes). It starts at address 0 and proceeds through address 1023. This provides 255 unique vectors; some of these are reserved for traps and other system functions. Of the 255, there are 192 reserved for user interrupt vectors. However, there is no protection on the first 64 entries, so the user interrupt vectors may overlap at the discretion of the systems designer.

Kinds of Exceptions — Exceptions can be generated by either internal or external causes. The externally generated exceptions are the interrupts and the bus error and reset requests. The interrupts are requests from peripheral devices for processor action while the bus error and reset inputs are used for access control and processor restart. The internally generated exceptions come from instructions, or

Table 6. REFERENCE CLASSIFICATION

Function Code Output			Reference Class
FC2	FC1	FC0	
0	0	0	(Unassigned)
0	0	1	User Data
0	1	0	User Program
0	1	1	(Unassigned)
1	0	0	(Unassigned)
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	Interrupt Acknowledge

Word 0	New Program Counter (High)	A0 = 0, A1 = 0
Word 1	New Program Counter (Low)	A0 = 0, A1 = 1

Figure 25. Exception Vector Format

D15 D8	D7							D0
Ignored	v7	v6	v5	v4	v3	v2	v1	v0

Where
v7 is the MSB of the Vector Number
v0 is the LSB of the Vector Number

Figure 26. Peripheral Vector Number Format

A23 A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
All Zeroes	v7	v6	v5	v4	v3	v2	v1	v0	0	0

Figure 27. Address Translated from 8-Bit Vector Number

from address errors or tracing. The trap (TRAP), trap on overflow (TRAPV), check register against bounds (CHK) and divide (DIV) instructions all can generate exceptions as part of their instruction execution. In addition, illegal instructions, word fetches from odd addresses and privilege violations cause exceptions. Tracing behaves like a very high priority, internally generated interrupt after each instruction execution.

Exception Processing Sequence — Exception processing occurs in four identifiable steps. In the first step, an internal copy is made of the status register. After the copy is made, the S-bit is asserted putting the processor into the supervisor privilege state. Also, the T-bit is negated which will allow the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor fetch, classified as an interrupt acknowledge. For all other exceptions, internal logic provides the vectored number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current processor status, except in the case of the reset exception. The current program counter value and the saved copy of the status register are stacked using the supervisor stack pointer. The program counter value stacked usually points to the next unexecuted instruction, however for bus error and address error, the value stacked for the program counter is unpredictable, and may be incremented from the address of the instruction which caused the error. Additional information defining the current context is stacked for the bus error and address error exceptions.

The last step is the same for all exceptions. The new program counter value is fetched from the exception vector and the processor resumes instruction execution. The instruction at the address given in the exception vector is fetched, and the normal instruction decoding and execution is started.

Multiple Exceptions — The following describes the processing which occurs when multiple exceptions arise simultaneously. Exceptions can be grouped according to their occurrence and priority. The group 0 exceptions are reset, bus error, and address error. These exceptions cause the instruction currently being executed to be aborted and the exception processing to commence at the next minor cycle of the processor. The group 1 exceptions are trace and interrupt, as well as the privilege violations and illegal instructions. These exceptions allow the current instruction to execute to completion, but preempt the execution of the next instruction by forcing exception processing to occur (privilege violations and illegal instructions are detected when they are the next instruction to be executed). The group 2 exceptions occur as part of the

Table 7. EXCEPTION VECTOR ASSIGNMENT

Vector Number(s)	Address			Assignment
	Dec	Hex	Space	
0	0	000	SP	Reset Initial SSP
–	4	004	SP	Reset Initial PC
2	8	008	SD	Bus Error
3	12	00C	SD	Address Error
4	16	010	SD	Illegal Instruction
5	20	014	SD	Zero Divide
6	24	018	SD	CHK Instruction
7	28	01C	SD	TRAPV Instruction
8	32	020	SD	Privilege Violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 Emulator
11	44	02C	SD	Line 1111 Emulator
12*	48	030	SD	(Unassigned, reserved)
13*	52	034	SD	(Unassigned, reserved)
14*	56	038	SD	(Unassigned, reserved)
15	60	03C	SD	Uninitialized Interrupt Vector
16-23*	64	04C	SD	(Unassigned, reserved)
	95	06F		–
24	96	060	SD	Spurious Interrupt
25	100	064	SD	Level 1 Interrupt Autovector
26	104	068	SD	Level 2 Interrupt Autovector
27	108	06C	SD	Level 3 Interrupt Autovector
28	112	070	SD	Level 4 Interrupt Autovector
29	116	074	SD	Level 5 Interrupt Autovector
30	120	078	SD	Level 6 Interrupt Autovector
31	124	07C	SD	Level 7 Interrupt Autovector
32-47	128	080	SD	TRAP Instruction Vectors
	191	0BF		–
48-63*	192	0C0	SD	(Unassigned, reserved)
	255	0FF		–
64-255	256	100	SD	User Interrupt Vectors
	1023	3FF		–

*Vector numbers 12, 13, 14, 16 through 23 and 48 through 63 are reserved for future enhancements. No user peripheral devices should be assigned these numbers.

normal processing of instructions. The TRAP, TRAPV, CHK, and zero divide exceptions are in this group. For these exceptions, the normal execution of an instruction may lead to exception processing.

Group 0 exceptions have highest priority, while group 2 exceptions have lowest priority. Within group 0, reset has highest priority, followed by bus error and then address error. Within group 1, trace has priority over external interrupts, which in turn takes priority over illegal instruction and privilege violation. Since only one instruction can be executed at a time, there is no priority relation within group 2.

The priority relation between two exceptions determines which is taken, or taken first, if the conditions for both arise simultaneously. Therefore, if a bus error occurs during a TRAP instruction, the bus error takes precedence, and the TRAP instruction processing is aborted. In another example, if an interrupt request occurs during the execution of an instruction while the T-bit is asserted, the trace exception has priority, and is processed first. Before instruction processing resumes, however, the interrupt exception is also processed, and instruction processing commences finally in the interrupt handler routine. A summary of exception grouping and priority is given in table 8.

Exception Processing Detailed Discussion

Exceptions have a number of sources, and each exception has processing which is peculiar to it.

Reset — The reset input provides the highest exception level. The processing of the reset signal is designed for system initiation and recovery from catastrophic failure. Any processing in progress at the time of the reset is aborted and cannot be recovered. The processor is forced into the supervisor state, and the trace state is forced off. The processor interrupt priority mask is set at level seven. The vector number is internally generated to reference the reset exception vector at location 0 in the supervisor program space. Because no assumptions can be made about the validity of register contents, in particular the supervisor stack pointer, neither the program counter nor the status register is saved. The address contained in the first two words of the reset exception vector is fetched as the initial supervisor stack pointer, and the address in the last two words of the reset exception vector is fetched as the initial program counter. Finally, nstruction execution is started at the address in the program counter. The power-up/restart code should be pointed to by the initial program counter.

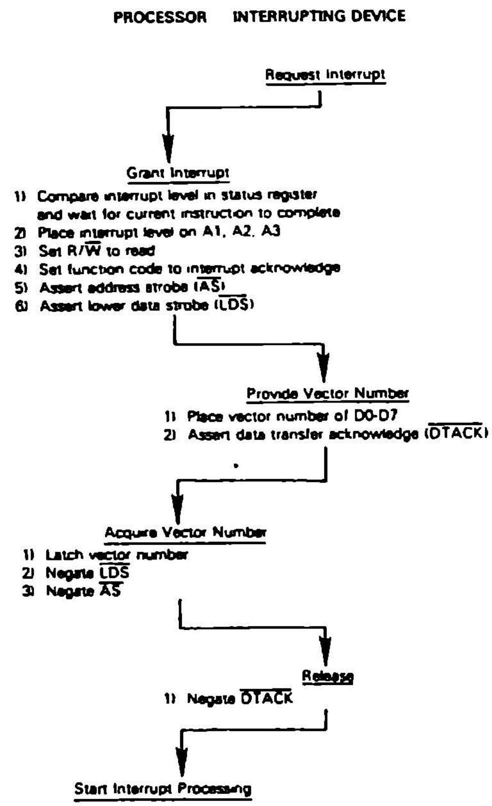
The RESET instruction does not cause loading of the reset vector, but does assert the reset line to reset external devices. This allows the software to reset the system to a known state and then continue processing with the next instruction.

Interrupts — Seven levels of interrupt priorities are provided. Devices may be chained externally within interrupt priority levels, allowing an unlimited number of peripheral devices to interrupt the processor. Interrupt priority levels are numbered from one to seven, level seven being the highest priority. The status register contains a three-bit mask which indicates the current processor priority, and interrupts are inhibited for all priority levels less than or equal to the current processor priority.

An interrupt request is made to the processor by encoding the interrupt request level on the interrupt request lines; a zero

Table 8. EXCEPTION GROUPING AND PRIORITY

Group	Exception	Processing
0	Reset Bus Error Address Error	Exception processing begins within two clock cycles
1	Trace Interrupt Illegal Privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV, CHK, Zero Divide	Exception processing is started by normal instruction execution



Request Interrupt

Grant Interrupt
1) Compare interrupt level in status register and wait for current instruction to complete
2) Place interrupt level on A1, A2, A3
3) Set R/$\overline{W}$ to read
4) Set function code to interrupt acknowledge
5) Assert address strobe ($\overline{AS}$)
6) Assert lower data strobe ($\overline{LDS}$)

Provide Vector Number
1) Place vector number of D0-D7
2) Assert data transfer acknowledge ($\overline{DTACK}$)

Acquire Vector Number
1) Latch vector number
2) Negate $\overline{LDS}$
3) Negate $\overline{AS}$

Release
1) Negate $\overline{DTACK}$

Start Interrupt Processing

Figure 28. Interrupt Acknowledge Sequence Flow Chart

indicates no interrupt request. Interrupt requests arriving at the processor do not force immediate exception processing but are made pending. Pending interrupts are detected between instruction executions. If the priority of the pending interrupt is lower than or equal to the current processor priority, execution continues with the next instruction and the interrupt exception processing is postponed. (The recognition of level seven is slighly different, as explained in a following paragraph.)

If the priority of the pending interrupt is greater than the current processor priority, the exception processing sequence is started. First a copy of the status register is saved, and the privilege state is set to supervisor, tracing is suppressed, and the processor priority level is set to the level of the interrupt being acknowledged. The processor fetches the vector number from the interrupting device, classifying the reference as an interrupt acknowledge and displaying the level number of the interrupt being acknowledged on the address bus. If external logic requests an automatic vectoring, the processor internally generates a vector number which is determined by the interrupt level number. If external logic indicates a bus error, the interrupt is taken to be spurious, and the generated vector number references the spurious interrupt vector. The processor then proceeds with the usual exception processing, saving the program counter and status register on the supervisor stack. The saved value of the program counter is the address of the instruction which would have been executed had the interrupt not been present. The content of the interrupt vector whose vector number was previously obtained is fetched and loaded into the program counter, and normal instruction execution commences in the interrupt handling routine. A flow chart for the interrupt acknowledge sequence is given in figure 28, a timing diagram is given in figure 29 and the interrupt exception timing sequence is shown in figure 30.

Priority level seven is a special case. Level seven interrupts cannot be inhibited by the interrupt priority mask, thus providing a 'non-maskable interrupt' capability. An interrupt is generated each time the interrupt request level changes from some lower level to level seven. Note that a level seven interrupt may still be caused by the level comparison if the request level is a seven and the processor is set to a lower level by an instruction.

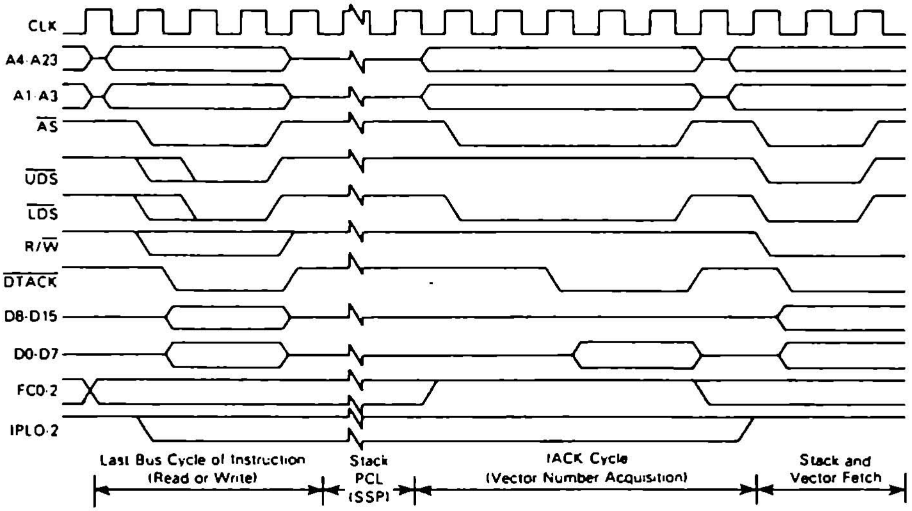

CLK
A4-A23
A1-A3
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
FC0-2
IPL0-2
Last Bus Cycle of Instruction (Read or Write)
Stack PCL (SSP)
IACK Cycle (Vector Number Acquisition)
Stack and Vector Fetch

Figure 29. Interrupt Acknowledge Sequence Timing Diagram

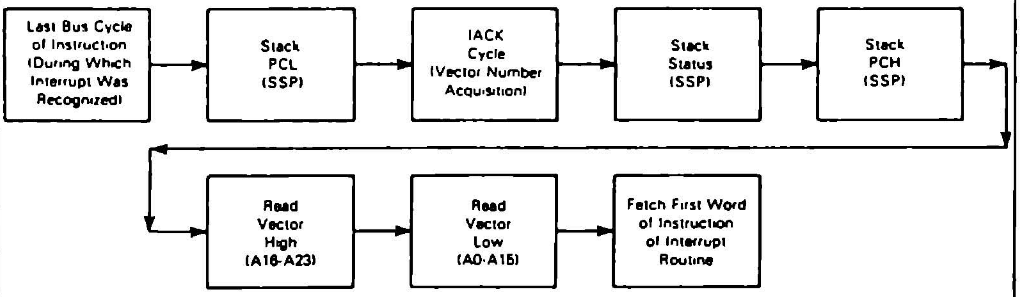

Last Bus Cycle of Instruction (During Which Interrupt Was Recognized)
Stack PCL (SSP)
IACK Cycle (Vector Number Acquisition)
Stack Status (SSP)
Stack PCH (SSP)
Read Vector High (A16-A23)
Read Vector Low (A0-A15)
Fetch First Word of Instruction of Interrupt Routine

Figure 30. Interrupt Exception Timing Sequence

Uninitialized Interrupt — An interrupting device asserts VPA or provides an interrupt vector during an interrupt acknowledge cycle to the SC68000. If the vector register has not been initialized, the responding SC68000 family peripheral will provide vector 15, the uninitialized interrupt vector. This provides a uniform way to recover from a programming error.

Spurious Interrupt — If during the interrupt acknowledge cycle no device responds by asserting DTACK or VPA, the bus error line should be asserted to terminate the vector acquisition. The processor separates the processing of this error from the bus error by fetching the spurious interrupt vector instead of the bus error vector. The processor then proceeds with the usual exception processing.

Instruction Traps — Traps are exceptions caused by instructions. They arise either from processor recognition of abnormal conditions during instruction execution, or from use of instructions whose normal behavior is trapping.

Some instructions are used specifically to generate traps. The TRAP instruction always forces an exception, and is useful for implementing system calls for user programs. The TRAPV and CHK instructions force an exception if the user program detects a runtime error, which may be an arithmetic overflow or a subscript out of bounds. The signed divide (DIVS) and unsigned divide (DIVU) instructions will force an exception if a division operation is attempted with a divisor of zero.

Illegal and Unimplemented Instructions — Illegal instruction is the term used to refer to any of the word bit patterns which are not the bit pattern of the first word of a legal instruction. During instruction execution, if such an instruction is fetched, an illegal instruction exception occurs.

Word patterns with bits 15 through 12 equaling 1010 or 1111 are distinguished as unimplemented instructions and separate exception vectors are given to these patterns to permit efficient emulation. This facility allows the operating system to detect program errors, or to emulate unimplemented instructions in software.

Privilege Violations — In order to provide system security, various instructions are privileged. An attempt to execute one of the privileged instructions while in the user state will cause an exception. The privileged instructions are:

STOP
RESET
RTE
MOVE to SR
AND (word) immediate to SR
EOR (word) immediate to SR
OR (word) immediate to SR
MOVE USP

Tracing — To aid in program development, the SC68000 includes a facility to allow instruction by instruction tracing. In the trace state, after each instruction is executed, an exception is forced allowing a debugging program to monitor the execution of the program under test.

The trace facility uses the T-bit in the supervisor portion of the status register. If the T-bit is negated (off), tracing is disabled and instruction execution proceeds from instruction to instruction as normal. If the T-bit is asserted (on) at the beginning of the execution of an instruction, a trace exception will be generated after the execution of that instruction is completed. If the instruction is not executed, either because an interrupt is taken, or the instruction is illegal or privileged, the trace exception does not occur. The trace exception also does not occur if the instruction is aborted by a reset, bus error, or address error exception. If the instruction is indeed executed and an interrupt is pending on completion, the trace exception is processed before the interrupt exception. If, during the execution of the instruction, an exception is forced by that instruction, the forced exception is processed before the trace exception.

As an extreme illustration of the above rules, consider the arrival of an interrupt during the execution of a TRAP instruction while tracing is enabled. First the trap exception is processed, then the trace exception, and finally the interrupt exception. Instruction execution resumes in the interrupt handler routine.

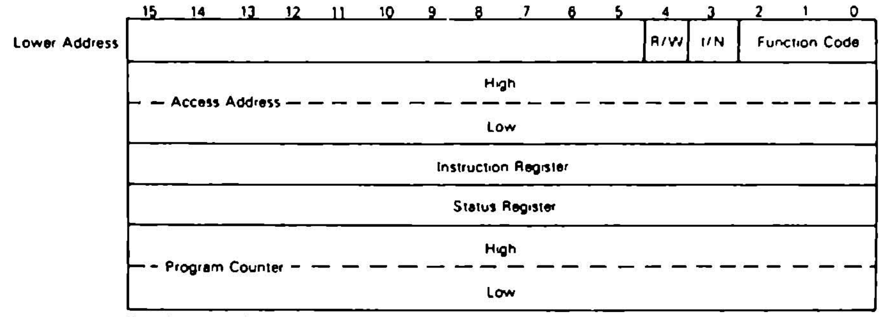
Bus Error — Bus error exceptions occur when external logic requests that a bus error be processed by an exception. The current bus cycle which the processor is making is then aborted. Whether the processor was doing instruction or exception processing, that processing is terminated, and the processor immediately begins exception processing.

Exception processing for bus error follows the usual sequence of steps. The status register is copied, the supervisor state is entered, and the trace state is turned off. The vector number is generated to refer to the bus error vector. Since the processor was not between instructions when the bus error exception request was made, the context of the processor is more detailed. To save more of this context, additional information is saved on the supervisor stack. The program counter and the copy of the status register are of course saved. The value saved for the program counter is advanced by some amount, two to ten bytes beyond the address of the first word of the instruction which made the reference causing the bus error. If the bus error occurred during the fetch of next instruction, the saved program counter has a value in the vicinity of the current instruction, even if the current instruction is a branch, a jump, or a return instruction. Besides the usual information, the processor saves its internal copy of the first word of the instruction being processed, and the address which was being accessed by the aborted bus cycle. Specific information about the access is also saved: whether it was a read or a write, whether the processor was processing an instruction or not, and the classification displayed on the function code outputs when the bus error occurred. The processor is processing an instruction if it is in the normal state or processing a group 2 exception; the processor is not processing an instruction if it is processing a group 0 or group 1 exception. Figure 31 illustrates how this information is organized on the supervisor stack. Although this information is not sufficient in general to effect full recovery from the bus error, it does allow software diagnosis. Finally, the processor commences instruction processing at the address contained in the vector. It is the responsibility of the error handler routine to clean up the stack and determine where to continue execution.

If a bus error occurs during the exception processing for a bus error, address error, or reset, the processor is halted, and all processing ceases. This simplifies the detection of catastrophic system failure, since the processor removes itself from the system rather than destroying all memory contents. Only the RESET pin can restart a halted processor.

Address Error

Address error exceptions occur when the processor attempts to access a word or a long word operand or an instruction at an odd address. The effect is much like an internally generated bus error, so that the bus cycle is aborted, and the processor ceases whatever processing it is currently doing and begins exception processing.


15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Lower Address
R/W
I/N
Function Code
High
Access Address
Low
Instruction Register
Status Register
High
Program Counter
Low

R/W (read/write) write = 0, read = 1 I/N (instruction/not) instruction = 0, not = 1

Figure 31. Supervisor Stack Order

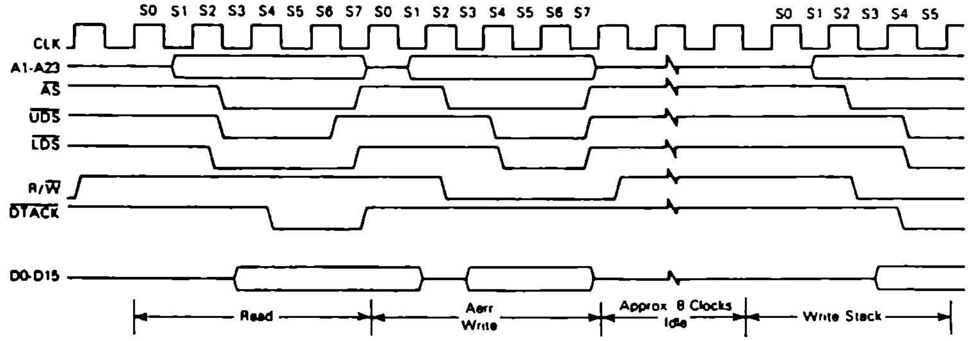

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
S0 S1 S2 S3 S4 S5
CLK
A1-A23
AS
UDS
LDS
R/W
DTACK
D0-D15
Read
Aerr Write
Approx 8 Clocks Idle
Write Stack

Figure 32. Address Error Timing

After exception processing commences, the sequence is the same as that for bus error including the information that is stacked, except that the vector number refers to the address error vector instead. Likewise, if an address error occurs during the exception processing for a bus error, address error, or reset, the processor is halted. As show in figure 32, an address error will execute a short bus cycle followed by exception processing.

INTERFACE WITH SYNCHRONOUS PERIPHERALS

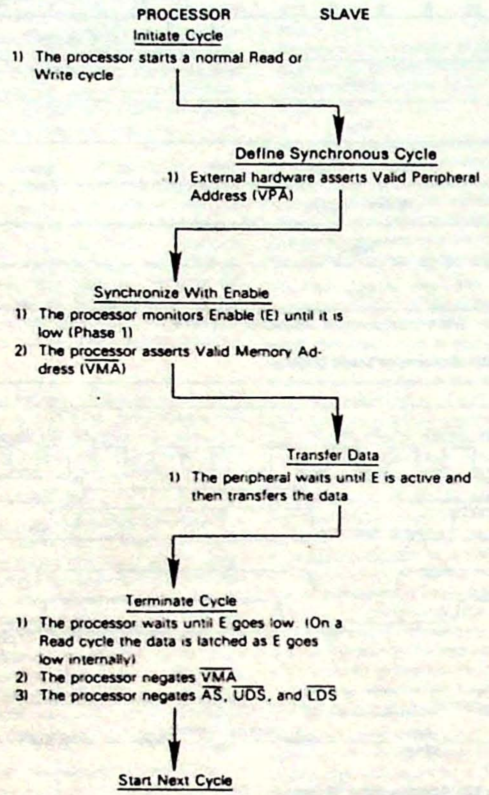
To interface synchronous peripherals with the asynchronous SC68000, the processor modifies its bus cycle to meet the synchronous cycle requirements whenever a synchronous device address is detected. Figure 33 is a flow chart of the interface operation between the processor and synchronous devices.

Data Transfer Operation

Three signals on the processor provide the synchronous interface. They are: enable (E), valid memory address (VMA), and valid peripheral address (VPA). The bus frequency is one tenth of the incoming SC68000 clock frequency. Enable has a 60/40 duty cycle; that is, it is low for six input clocks and high for four input clocks. This duty cycle allows the processor to do successive VPA accesses on successive E pulses.

The synchronous cycle timing diagrams and corresponding AC electrical characteristics table are located towards the end of this data sheet. At state zero (S0) in the cycle, the address bus and function codes are in the high-impedance state. One half clock later, in state 1, the address bus and function code outputs are released from the high-impedance state.

During state 2, the address strobe (AS) is asserted to indicate that there is a valid address on the address bus. If the bus cycle is a read cycle, the upper and/or lower data strobes are also asserted in state 2. If the bus cycle is a write cycle, the read/write (R/W) signal is switched to low (write) during state 2. One half clock later, in state 3, the write data is placed on the



Initiate Cycle

1) The processor starts a normal Read or Write cycle

Define Synchronous Cycle

1) External hardware asserts Valid Peripheral Address ($\overline{VPA}$)

Synchronize With Enable

1) The processor monitors Enable (E) until it is low (Phase 1)
2) The processor asserts Valid Memory Address ($\overline{VMA}$)

Transfer Data

1) The peripheral waits until E is active and then transfers the data

Terminate Cycle

1) The processor waits until E goes low. (On a Read cycle the data is latched as E goes low internally)
2) The processor negates $\overline{VMA}$
3) The processor negates $\overline{AS}$, $\overline{UDS}$, and $\overline{LDS}$

Start Next Cycle

Figure 33. Synchronous Interfacing Flow Chart

data bus, and in state 4 the data strobes are issued to indicate valid data on the data bus.

The processor now inserts wait states until it recognizes the assertion of VPA. The VPA input signals the processor that the address on the bus is the address of a synchronous device (or an area reserved for synchronous devices) and that the bus should conform to the synchronous transfer characteristics of the synchronous bus. Valid peripheral address is derived by decoding the address bus, conditioned by the address strobe.

After the recognition of VPA, the processor assures that enable (E) is low, by waiting if necessary, and subsequently asserts VMA. Valid memory address is then used as part of the chip select equation of the peripheral. This ensures that the synchronous peripherals are selected and deselected at the correct time. The peripheral now runs its cycle during the high portion of the E signal. Cycle timing diagrams depicting best and worst cases are located towards the end of this data sheet. The cycle length is dependent strictly on when VPA is asserted in relationship to the E clock.

During a read cycle, the processor latches the peripheral data in state 6. For all cycles, the processor negates the address and data strobes one half clock cycle later in state 7, and the enable signal goes low at this time. Another half clock later, the address bus is put in the high-impedance state. During a write cycle, the data bus is put in the high-impedance state and the read/write signal is switched high at this time. The peripheral logic must remove VPA within one clock after address strobe is negated.

DTACK should not be asserted while VPA is asserted. The SC68000 VMA is active low, while the VMA of the synchronous device should be active high. This allows the processor to put its buses in the high-impedance state on DMA requests without inadvertently selecting peripherals.

Interrupt Operation

During an interrupt acknowledge cycle while the processor is fetching the vector, if VPA is asserted, the SC68000 will assert VMA and complete a synchronous read cycle as shown in figure 34. The processor will then use an internally generated vector that is a function of the interrupt being serviced. This process is known as autovectoring. The seven autovectors are vector numbers 25 through 31 (decimal).

There are six normal interrupt vectors and one NMI type vector. As with the SC68000's normal vectored interrupt, the interrupt service routine can be located anywhere in the address space. This is due to the fact that while the vector numbers are fixed, the contents of the vector table entries are assigned by the user.

Since VMA is asserted during autovectoring, the synchronous peripheral address decoding should prevent unintended accesses.

DATA TYPES AND ADDRESSING MODES

Five basic data types are supported:

- Bits
- BCD digits (8-bits)
- Bytes (8-bits)
- Word (16-bits)
- Long words (32-bits)

In addition, operations on other data types such as memory addresses, status word data, etc. are provided for in the instruction set. The 14 addressing modes (see table 9) include six basic types.

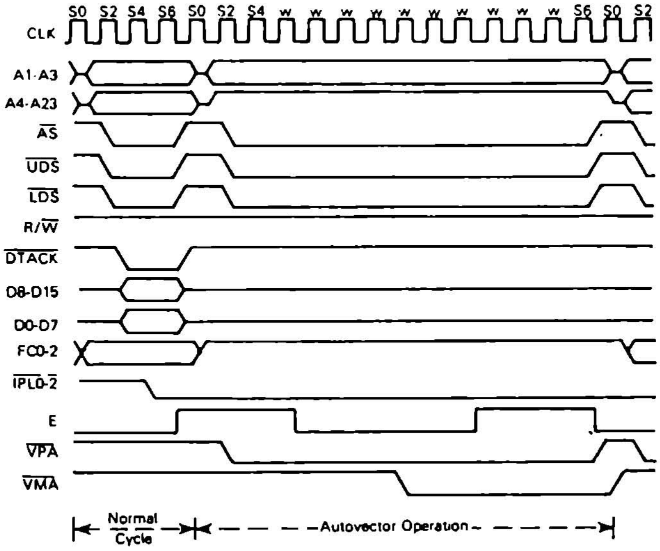


Figure 34. Autovector Operation Timing

- Register direct
- Register indirect
- Absolute
- Immediate
- Program counter relative
- Implied

Included in the register indirect addressing modes is the capability to do post-incrementing, pre-decrementing, offsetting and indexing. Program counter relative mode can also be modified via indexing and offsetting.

Instruction Format

Instructions are from one to five words in length, a shown in figure 35. The length of the instruction and the operation to be performed are specified by the first word of the instruction which is called the operation word. The remaining words further specify the operands. These words are either immediate operands or extensions to the effective address modes specified in the operation word.

Program/Data References

The SC68000 separates memory references into two classes: program references and data references. Program references, as the name implies, are references to that section of memory that contains the program being executed. Data references refer to that section of memory that contains data. Generally, operand reads are from the data space. All operand writes are to the data space.

Addressing

Instructions for the SC68000 contain two kinds of information; the type of function to be performed and the location of the operand(s) on which to perform the function. Instructions specify an operand location in one of three ways:

Register specification — the number of the register is given in the register field of the instruction.

Effective address — use of the different effective address modes.

Implicit reference — the definition of certain instructions implies the use of specific registers.

Register Specification

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register selected is an address or data register and how the register is to be used.

Effective Address

Most instructions specify the location of an operand by using the effective address field in the operation word. For example, figure 36 shows the general format of the single effective address instruction operation word. The effective address is composed of two 3-bit fields: the mode field, and the register field. The value in the mode field selects the different address modes. The register field contains the number of a register.

Table 9. DATA ADDRESSING MODES

Mode	Generation
Register Direct Addressing	
Data Register Direct	EA = Dn
Address Register Direct	EA = An
Absolute Data Addressing	
Absolute Short	EA = (Next Word)
Absolute Long	EA = (Next Two Words)
Program Counter Relative Addressing	
Relative with Offset	EA = (PC) + d_{16}
Relative with Index and Offset	EA = (PC) + (Xn) + d_8
Register Indirect Addressing	
Register Indirect	EA = (An)
Postincrement Register Indirect	EA = (An), An ← An + N
Predecrement Register Indirect	An ← An – N, EA = (An)
Register Indirect with Offset	EA = (An) + d_{16}
Indexed Register Indirect with Offset	EA = (An) + (Xn) + d_8
Immediate Data Addressing	
Immediate	DATA = Next Word(s)
Quick Immediate	Inherent Data
Implied Addressing	
Implied Register	EA = SR, USP, SP, PC

NOTES:

EA = Effective Address
An = Address Register
Dn = Data Register
Xn = Address or Data Register used as Index Register
SR = Status Register
PC = Program Counter
( ) = Contents of

d_8 = Eight-bit Offset (displacement)
d_{16} = Sixteen-bit Offset (displacement)
N = 1 for Byte, 2 for Words and 4 for Long Words
← = Replaces

The effective address field may require additional information to fully specify the operand. This additional information, called the effective address extension, is contained in a following word or words and is considered part of the instruction, as shown in figure 35. The effective address modes are grouped into three categories: register direct, memory addressing, and special.

Register Direct Modes

These effective addressing modes specify that the operand is in one of the 16 multifunction registers.

Data Register Direct — The operand is in the data register specified by the effective address register field.

Address Register Direct — The operand is in the address register specified by the effective address register field.

Memory Address Modes

These effective addressing modes specify that the operand is in memory and provide the specific address of the operand.

Address Register Indirect — The address of the operand is in the address register specified by the register field. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation Word (First Word Specifies Operation and Modes)															
Immediate Operand (If Any, One or Two Words)															
Source Effective Address Extension (If Any, One or Two Words)															
Destination Effective Address Extension (If Any, One or Two Words)															

Figure 35. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X	X	X	Effective Address Mode			Register		

Figure 36. Single Effective Address Instruction Operation Word General Format

Address Register Indirect with Postincrement — The address of the operand is in the address register specified by the register field. After the operand address is used, it is incremented by one, two, or four depending on whether the size of the operand is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is incremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Predecrement — The address of the operand is in the address register specified by the register field. Before the operand and the address is used, it is decremented by one, two or four depending on whether the operand size is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is decremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Displacement — This address mode requires one word of extension. The address of the operand is the sum of the address in the address register and the sign-extended 16-bit displacement integer in the extension word. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

Address Register Indirect with Index — This address mode requires one word of extension. The address of the operand is the sum of the address in the address register, the sign-extended displacement integer in the low order eight bits of the extension word, and the contents of the index register. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

Special Addressing Modes

The special address modes use the effective address register field to specify the special addressing mode instead of a register number.

Absolute Short Address — This address mode requires one word of extension. The address of the operand is in the extension word. The 16-bit address is sign extended before it is used. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

Absolute Long Address — This address mode requires two words of extension. The address of the operand is developed by the concatenation of the extension words. The high-order part of the address is the first extension word; the low order part of the address is the second extension word. The reference is classified as a data reference with the exception of the jump and the jump to subroutine instructions.

Program Counter with Displacement — This address mode requires one word of extension. The address of the operand is the sum of the address in the program counter and the sign-extended 16-bit displacement integer in the extension word. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

Program Counter with Index — This address mode requires one word of extension. The address is the sum of the address in the program counter, the sign-extended displacement integer in the lower eight bits of the extension word, and the contents of the index register. The value in the program counter is the address of the extension word. This reference is classifed as a program reference.

Immediate Data — This address mode requires either one or two words of extension depending on the size of the operation.

Byte operation — operand is low order byte of extension word.

Word operation — operand is extension word.

Long word operation — operand is in the two extension words, high-order 16 bits are in the first extension word, low-order 16 bits are in the second extension word.

Condition Codes or Status Register — A selected set of instructions may reference the status register by means of the effective address field. These are:

ANDI to CCR
ANDI to SR
EORI to SR
ORI to CCR
ORI to SR

Effective Address Encoding Summary

Table 10 is a summary of the effective addressing modes.

Implicit Reference

Some instructions make implicit reference to the program counter (PC), the system stack pointer (SP), the supervisor stack pointer (SSP), the user stack pointer (USP), or the status register (SR).

System Stack

The system stack is used implicitly by many instructions; user stacks and

Table 10. EFFECTIVE ADDRESS ENCODING SUMMARY

Addressing Mode	Mode	Register
Data Register Direct	000	register number
Address Register Direct	001	register number
Address Register Indirect	010	register number
Address Register Indirect with Postincrement	011	register number
Address Register Indirect with Predecrement	100	register number
Address Register Indirect with Displacement	101	register number
Address Register Indirect with Index	110	register number
Absolute Short	111	000
Absolute Long	111	001
Program Counter with Displacement	111	010
Program Counter with Index	111	011
Immediate	111	100

queues may be created and maintained through the addressing modes. Address register seven (A7) is the stack pointer (SP). The SP is either the SSP or the USP, depending on the state of the S-bit in the status register. If the S-bit indicates supervisor state, SSP is the active system stack pointer, and the USP cannot be referenced as an address register. If the S-bit indicates user state, the USP is the active system stack pointer, and the SSP cannot be referenced. Each system stack fills from high memory to low memory.

INSTRUCTION SET OVERVIEW

The SC68000 instruction set is summarized in table 11. Some additional instructions are variations, or subsets, of these and appear in table 12. Special emphasis has been given to the instruction set's support of structured high-level languages to facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words, and most instructions can use any of the 14 addressing modes. Combining instruction types, data types, and addressing modes over 1000 useful instructions are provided These instructions include signed and unsigned multiply and divide, 'quick' arithmetic operations, BCD arithmetic and expanded operations (through traps).

The instructions form a set of tools that include all machine functions to perform the following operations:

Data movement
Integer arithmetic

Table 11. INSTRUCTION SET SUMMARY

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
Bcc	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DBcc	Test Condition, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

Table 12. VARIATIONS OF INSTRUCTION TYPES

Instruction Type	Variation	Description
ADD	ADD	Add
	ADDA	Add Address
	ADDQ	Add Quick
	ADDI	Add Immediate
	ADDX	Add with Extend
AND	AND	Logical And
	ANDI	And Immediate
CMP	CMP	Compare
	CMPA	Compare Address
	CMPM	Compare Memory
	CMPI	Compare Immediate
EOR	EOR	Exclusive Or
	EORI	Exclusive Or Immediate
MOVE	MOVE	Move
	MOVEA	Move Address
	MOVEQ	Move Quick
	MOVE from SR	Move from Status Register
	MOVE to SR	Move to Status Register
	MOVE to CCR	Move to Condition Codes
	MOVE USP	Move User Stack Pointer
NEG	NEG	Negate
	NEGX	Negate with Extend
OR	OR	Logical Or
	ORI	Or Immediate
SUB	SUB	Subtract
	SUBA	Subtract Address
	SUBI	Subtract Immediate
	SUBQ	Subtract Quick
	SUBX	Subtract with Extend

- Logical
- Shift and rotate
- Bit manipulation
- Binary coded decimal
- Program control
- System control

The complete range of instruction capabilities, combined with the variety of addressing modes described previously, provide a very flexible base for program development.

Data Movement Operations

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data move instructions allow byte, word, and long word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address move instructions allow word and long word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction, there are several special data movement instructions: move multiple register (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), and move quick (MOVEQ). Table 13 is a summary of the data movement operations.

Table 13. DATA MOVEMENT OPERATIONS

Instruction	Operand Size	Operation
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	–	An → SP@ - SP → An SP + d → SP
MOVE	8, 16, 32	(EA)s → EAd
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVEP	16, 32	(EA) → Dn Dn → EA
MOVEQ	8	#xxx → Dn
PEA	32	EA → SP@ -
SWAP	32	Dn[31:16] ↔ Dn[15:0]
UNLK	–	An → Sp SP@ + → An

NOTES
s = source
d = destination
[ ] = bit numbers
@ – = indirect with predecrement
@ + = indirect with post increment

Integer Arithmetic Operations

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV) as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions can be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long word product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extended (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 14 is a summary of the integer arithmetic operations.

Logical Operations

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 15 is a summary of the logical operations.

Shift and Rotate Operations

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand sizes and allow a shift count specified in the instruction of one to eight bits, or 0 to 63 specified in a data register. Memory shifts and rotates are for word operands only and allow only single-bit shifts or rotates. Table 16 is a summary of the shift and rotate operations.

Bit Manipulation Operations

Bit manipulation operations are accomplished using bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 17 is a summary of the bit manipulation operations (bit 2 of the status register is Z).

Binary Coded Decimal Operations

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 18 is a summary of the binary coded decimal operations.

Program Control Operations

Program control operations are accomplished using a series of conditional and unconditional branch instructions and return instructions (see table 19). The conditional instructions provide setting and branching for the following conditions:

CC—carry clear
CS—carry set
EQ—equal
F—never true
GE—greater or equal
GT—greater than
HI—high
LE—less or equal
LS—low or same

- LT—less than
- MI—minus
- NE—not equal
- PL—plus
- T—always true
- VC—no overflow
- VS—overflow

System Control Operations

System control operations are accomplished by using priviledged instructions, trap generating instructions, and instructions that use or modify the status register. These instructions are summarized in table 20.

INSTRUCTION SET

The following provides information about the addressing categories and instructior set of the SC68000.

Addressing Categories

Effective address modes can be categorized by the ways in which they may be

Table 14. INTEGER ARITHMETIC OPERATIONS

Instruction	Operand Size	Operation
ADD	8, 16, 32 16, 32	Dn + (EA) → Dn (EA) + Dn → EA (EA) + #xxx → EA An + (EA) → An
ADDX	8, 16, 32 16, 32	Dx + Dy + X → Dx Ax@ – Ay@ – + X → Ax@
CLR	8, 16, 32	0 → EA
CMP	8, 16, 32 16, 32	Dn – (EA) (EA) – #xxx Ax@ + – Ay@ + An – (EA)
DIVS	32 ÷ 16	Dn/(EA) → Dn
DIVU	32 ÷ 16	Dn/(EA) → Dn
EXT	8 → 16 16 → 32	$(Dn)_8 \rightarrow Dn_{16}$ $(Dn)_{16} \rightarrow Dn_{32}$
MULS	16*16 → 32	Dn*(EA) → Dn
MULU	16*16 → 32	Dn*(EA) → Dn
NEG	8, 16, 32	0 – (EA) → EA
NEGX	8, 16, 32	0 – (EA) – X – EA
SUB	8, 16, 32 16, 32	Dn – (EA) → Dn (EA) – Dn → EA (EA) – #xxx → EA An – (EA) → An
SUBX	8, 16, 32	Dx – Dy – X → Dx Ax@ – – Ay@ – – X → Ax@
TAS	8	(EA) – 0, 1 → EA[7]
TST	8, 16, 32	(EA) – 0

NOTE: [ ] = bit number

Table 15. LOGICAL OPERATIONS

Instruction	Operand Size	Operation
AND	8, 16, 32	Dn∧(EA) → Dn (EA)∧Dn → EA (EA)∧#xxx → EA
OR	8, 16, 32	Dn ∨ (EA) → Dn (EA) ∨ Dn → EA (EA) ∨ #xxx → EA
EOR	8, 16, 32	(EA) ● Dy → EA (EA) ● #xxx → EA
NOT	8, 16, 32	~(EA) → EA

NOTE ~ = invert

Table 16. SHIFT AND ROTATE OPERATIONS

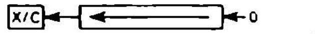
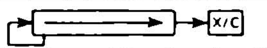
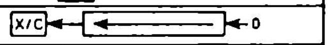
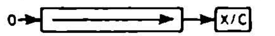
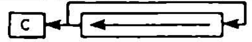
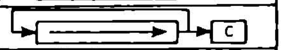
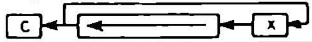
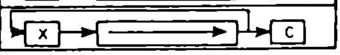
Instruction	Operand Size	Operation
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

Table 17. BIT MANIPULATION OPERATIONS

Instruction	Operand Size	Operation
BTST	8, 32	~ bit of (EA) → Z
BSET	8, 32	~ bit of (EA) → Z 1 → bit of EA
BCLR	8, 32	~ bit of (EA) → Z 0 → bit of EA
BCHG	8, 32	~ bit of (EA) → Z ~ bit of (EA) → bit of EA

Table 18. BINARY CODED DECIMAL OPERATIONS

Instruction	Operand Size	Operation
ABCD	8	$Dx_{10} + Dy_{10} + X \rightarrow Dx$ $Ax@ -_{10} + Ay@ -_{10} + X \rightarrow Ax@$
SBCD	8	$Dx_{10} - Dy_{10} - X \rightarrow Dx$ $Ax@ -_{10} - Ay@ -_{10} - X - Ax@$
NBCD	8	$0 - (EA)_{10} - X \rightarrow EA$

used. The following classifications are used in the instruction definitions:

Data	If an effective address mode may be used to refer to data operands, it is considered a data addressing effective address mode.
Memory	If an effective address mode may be used to refer to memory operands, it is considered a memory addressing effective address mode.
Alterable	If an effective address mode may be used to refer to alterable (writable) operands, it is considered an alterable addressing effective address mode.
Control	If an effective address mode may be used to refer to memory operands without an associated size, it is considered a control addressing effective address mode.

Table 21 shows the various categories to which each of the effective address modes belongs. Table 22 shows the instruction set.

The status register addressing mode is not permitted unless it is explicitly mentioned as a legal addressing mode.

These categories may be combined, so that additional, more restrictive, classifications may be defined. For example, the instruction descriptions use such classifications as alterable memory or data alterable. The former refers to those addressing modes which are both alterable and memory addresses, and the latter refers to addressing modes which are both data and alterable.

Instruction Prefetch

The SC68000 uses a two word tightly coupled instruction prefetch mechanism to enhance performance. This mechanism is described in terms of the microcode operations involved. If the execution of an instruction is defined to begin when the microroutine for that instruction is entered, some features of the prefetch mechanism can be described.

1. When execution of an instruction begins, the operation word and the word following have already been fetched. The operation word is in the instruction decoder.
2. In the case of multiword instructions, as each additional word of the instruction is used internally, a fetch is made to the instruction stream to replace it.
3. The last fetch from the instruction stream is made when the operation word is discarded and decoding is started on the next instruction.
4. If the instruction is a single word instruction causing a branch, the second word is not used. But because this word is fetched by the preceding instruction, it is impossible to avoid this superfluous fetch. In the case of an interrupt or trace exception, both words are not used.
5. The program counter usually points to the last word fetched from the instruction stream.

INSTRUCTION EXECUTION TIMES

The following contains listings of the instruction execution times in terms of the external clock (CLK) periods. In this timing data, it is also assumed that the memory cycle time is no greater than four periods

Table 19. PROGRAM CONTROL OPERATIONS

Instruction	Operation
Conditional	
B_{CC}	Branch conditionally (14 conditions) 8- and 16-bit displacement
DB_{CC}	Test condition, decrement, and branch 16-bit displacement
S_{CC}	Set byte conditionally (16 conditions)
Unconditional	
BRA	Branch always 8- and 16-bit displacement
BSR	Branch to subroutine 8- and 16-bit displacement
JMP	Jump
JSR	Jump to subroutine
Returns	
RTR	Return and restore condition codes
RTS	Return from subroutine

Table 20. SYSTEM CONTROL OPERATIONS

Instruction	Operation
Privileged	
RESET	Reset external devices
RTE	Return from exception
STOP	Stop program execution
ORI to SR	Logical OR to status register
MOVE USP	Move user stack pointer
ANDI to SR	Logical AND to status register
EORI to SR	Logical EOR to status register
MOVE EA to SR	Load new status register
Trap Generating	
TRAP	Trap
TRAPV	Trap on overflow
CHK	Check register against bounds
Status Register	
ANDI to CCR	Logical AND to condition codes
EORI to CCR	Logical EOR to condition codes
MOVE EA to CCR	Load new condition codes
ORI to CCR	Logical OR to condition codes
MOVE SR to EA	Store status register

Table 21. EFFECTIVE ADDRESSING MODE CATEGORIES

Effective Address Modes	Mode	Register	Data	Addressing Categories		
				Memory	Control	Alterable
Dn	000	register number	X	–	–	X
An	001	register number	–	–	–	X
An@	010	register number	X	X	X	X
An@ +	011	register number	X	X	–	X
An@ –	100	register number	X	X	–	X
An@(d)	101	register number	X	X	X	X
An@(d, ix)	110	register number	X	X	X	X
xxx W	111	000	X	X	X	X
xxx L	111	001	X	X	X	X
PC@(d)	111	010	X	X	X	–
PC@(d, ix)	111	011	X	X	X	–
#xxx	111	100	X	X	–	–

of the external processor clock input, which prevents the insertion of wait states in the bus cycle. The number of bus read and write cycles for each instruction is also included with the timing data. This data is enclosed in parenthesis following the execution periods and is shown as (r/w), where r is the number of read cycles and w is the number of write cycles. The number of periods includes instruction fetch and all applicable operand fetches and stores.

Effective Address Operand Calculation Timing

Table 23 lists the number of clock periods required to compute an instruction's effective address. It includes fetching of any extension words, the address computation, and fetching the memory operand. The number of bus read and write cycles is shown in parenthesis as (r/w). Note that there are no write cycles involved in processing the effective address.

Move Instruction Clock Periods

Table 24 and 25 indicate the number of clock periods for the move instruction. This data includes fetch, operand reads, and operand writes. The number of bus read and write cycles is shown in parenthesis as (r/w).

Standard Instruction Clock Periods

The number of clock periods shown in table 26 indicates the time required to perform the operations, store the results, and read the next instruction. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read cycles must be added to those of the effective address calculated where indicated.

In table 26, the headings have the following meanings: An = address register operand, Dn = data register operand, ea = an operand specified by an effective address, and M = memory effective address operand.

Immediate Instruction Clock Periods

The number of clock periods shown in table 27 includes the time to fetch immediate operands, perform the operations, store the results, and read the next operation. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

In table 27, the headings have the following meanings: # = immediate operand, Dn = data register operand, M = memory operand, and SR = status register.

Single Operand Instruction Clock Periods

Table 28 indicates the number of clock periods for the single operand instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Shift/Rotate Instruction Clock Periods

Table 29 indicates the number of clock periods for the shift and rotate instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Bit Manipulation Instruction Clock Periods

Table 30 indicates the number of clock periods for the bit manipulation instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Conditional Instruction Clock Periods

Table 31 indicates the number of clock periods required for the conditional instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

JMP, JSR, LEA, PEA, MOVEM Instruction Clock Periods

Table 32 indicates the number of clock periods required for the jump, jump to subroutine, load effective address, push effective address, and move multiple registers instructions. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 22. INSTRUCTION SET

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
ABCD	Add Decimal with Extend	(Destination)$_{10}$ + (Source)$_{10}$ → Destination	*	U	*	U	*
ADD	Add Binary	(Destination) + (Source) → Destination	*	*	*	*	*
ADDA	Add Address	(Destination) + (Source) → Destination	–	–	–	–	–
ADDI	Add Immediate	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDQ	Add Quick	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDX	Add Extended	(Destination) + (Source) + X → Destination	*	*	*	*	*
AND	AND Logical	(Destination) Λ (Source) → Destination	–	*	*	0	0
ANDI	AND Immediate	(Destination) Λ Immediate Data → Destination	–	*	*	0	0
ASL, ASR	Arithmetic Shift	(Destination) Shifted by <count> → Destination	*	*	*	*	*
B$_{CC}$	Branch Conditionally	If CC then PC + d → PC	–	–	–	–	–
BCHG	Test a Bit and Change	~(<bit number>) OF Destination → Z ~(<bit number>) OF Destination → <bit number> OF Destination	–	–	*	–	–
BCLR	Test a Bit and Clear	~(<bit number>) OF Destination → Z 0 → <bit number> → OF Destination	–	–	*	–	–
BRA	Branch Always	PC + d → PC	–	–	–	–	–
BSET	Test a Bit and Set	~(<bit number>) OF Destination → Z 1 → <bit number> OF Destination	–	–	*	–	–
BSR	Branch to Subroutine	PC → SP@ –, PC + d → PC	–	–	–	–	–
BTST	Test a Bit	~(<bit number>) OF Destination → Z	–	–	*	–	–
CHK	Check Register against Bounds	If Dn < 0 or Dn > (<ea>) then TRAP	–	*	U	U	U
CLR	Clear an Operand	0 → Destination	–	0	1	0	0
CMP	Compare	(Destination) – (Source)	–	*	*	*	*
CMPA	Compare Address	(Destination) – (Source)	–	*	*	*	*
CMPI	Compare Immediate	(Destination) – Immediate Data	–	*	*	*	*
CMPM	Compare Memory	(Destination) – (Source)	–	*	*	*	*
DB$_{CC}$	Test Condition, Decrement and Branch	If ~CC then Dn – 1 → Dn, if Dn ≠ – 1 then PC + d → PC	–	–	–	–	–
DIVS	Signed Divide	(Destination)/(Source) → Destination	–	*	*	*	0
DIVU	Unsigned Divide	(Destination)/(Source) → Destination	–	*	*	*	0
EOR	Exclusive OR Logical	(Destination) ⊕ (Source) → Destination	–	*	*	0	0
EORI	Exclusive OR Immediate	(Destination) ⊕ Immediate Data → Destination	–	*	*	0	0
EXG	Exchange Register	Rx ↔ Ry	–	–	–	–	–
EXT	Sign Extend	(Destination) Sign-extended → Destination	–	*	*	0	0
JMP	Jump	Destination → PC	–	–	–	–	–
JSR	Jump to Subroutine	PC → SP@ –; Destination → PC	–	–	–	–	–
LEA	Load Effective Address	Destination → An	–	–	–	–	–
LINK	Link and Allocate	An → SP@ –; SP → An; SP + d → SP	–	–	–	–	–
LSL, LSR	Logical Shift	(Destination) Shifted by <count> → Destination	*	*	*	0	*
MOVE	Move Data from Source to Destination	(Source) → Destination	–	*	*	0	0
MOVE to CCR	Move to Condition Code	(Source) → CCR	*	*	*	*	*
MOVE to SR	Move to the Status Register	(Source) → SR	*	*	*	*	*

* affected	0 cleared	U undefined
– unaffected	1 set	

Table 22. INSTRUCTION SET (Continued)

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
MOVE from SR	Move from the Status Register	SR → Destination	–	–	–	–	–
MOVE USP	Move User Stack Pointer	USP → An; An → USP	–	–	–	–	–
MOVEA	Move Address	(Source) → Destination	–	–	–	–	–
MOVEM	Move Multiple Registers	Registers → Destination (Source) → Registers	–	–	–	–	–
MOVEP	Move Peripheral Data	(Source) → Destination	–	–	–	–	–
MOVEQ	Move Quick	Immediate Data → Destination	–	•	•	0	0
MULS	Signed Multiply	(Destination)*(Source) → Destination	–	•	•	0	0
MULU	Unsigned Multiply	(Destination)*(Source) → Destination	–	•	•	0	0
NBCD	Negate Decimal with Extend	0 – (Destination)$_{10}$ – X → Destination	•	U	•	U	•
NEG	Negate	0 – (Destination) → Destination	•	•	•	•	•
NEGX	Negate with Extend	0 – (Destination) – X → Destination	•	•	•	•	•
NOP	No Operation	–	–	–	–	–	–
NOT	Logical Complement	~ (Destination) → Destination	–	•	•	0	0
OR	Inclusive OR Logical	(Destination) v (Source) → Destination	–	•	•	0	0
ORI	Inclusive OR Immediate	(Destination) v Immediate Data → Destination	–	•	•	0	0
PEA	Push Effective Address	Destination → SP@ –	–	–	–	–	–
RESET	Reset External Devices	–	–	–	–	–	–
ROL, ROR	Rotate (Without Extend)	(Destination) Rotated by <count> → Destination	–	•	•	0	•
ROXL, ROXR	Rotate with Extend	(Destination) Rotated by <count> → Destination	•	•	•	0	•
RTE	Return from Exception	SP@ + → SR; SP@ + → PC	•	•	•	•	•
RTR	Return and Restore Condition Codes	SP@ + → CC, SP@ + → PC	•	•	•	•	•
RTS	Return from Subroutine	SP@ + → PC	–	–	–	–	–
SBCD	Subtract Decimal with Extend	(Destination)$_{10}$ – (Source)$_{10}$ – X → Destination	•	U	•	U	•
S_{CC}	Set According to Condition	If $_{CC}$ then 1's → Destination else 0's → Destination	–	–	–	–	–
STOP	Load Status Register and Stop	Immediate Data → SR, STOP	•	•	•	•	•
SUB	Subtract Binary	(Destination) – (Source) → Destination	•	•	•	•	•
SUBA	Subtract Address	(Destination) – (Source) → Destination	–	–	–	–	–
SUBI	Subtract Immediate	(Destination) – Immediate Data → Destination	•	•	•	•	•
SUBQ	Subtract Quick	(Destination) – Immediate Data → Destination	•	•	•	•	•
SUBX	Subtract with Extend	(Destination) – (Source) – X → Destination	•	•	•	•	•
SWAP	Swap Register Halves	Register [31 16] → Register [15 0]	–	•	•	0	0
TAS	Test and Set an Operand	(Destination) Tested → CC; 1 → [7] OF Destination	–	•	•	0	0
TRAP	Trap	PC → SSP@ –, SR → SSP@ –, (Vector) → PC	–	–	–	–	–
TRAPV	Trap on Overflow	If V then TRAP	–	–	–	–	–
TST	Test an Operand	(Destination) Tested → CC	–	•	•	0	0
UNLK	Unlink	An → SP; SP@ + → An	–	–	–	–	–

[ ] = bit number

• affected   0 cleared   U undefined
– unaffected   1 set

Preliminary

Table 23. EFFECTIVE ADDRESS OPERAND CALCULATION TIMING

Addressing Mode		Byte, Word	Long
	Register		
Dn	Data Register Direct	0(0/0)	0(0/0)
An	Address Register Direct	0(0/0)	0(0/0)
	Memory		
An@	Address Register Indirect	4(1/0)	8(2/0)
An@ +	Address Register Indirect with Postincrement	4(1/0)	8(2/0)
An@ −	Address Register Indirect with Predecrement	6(1/0)	10(2/0)
An@(d)	Address Register Indirect with Displacement	8(2/0)	12(3/0)
An@(d, ix)*	Address Register Indirect with Index	10(2/0)	14(3/0)
xxx.W	Absolute Short	8(2/0)	12(3/0)
xxx.L	Absolute Long	12(3/0)	16(4/0)
PC@(d)	Program Counter with Displacement	8(2/0)	12(3/0)
PC@(d, ix)*	Program Counter with Index	10(2/0)	14(3/0)
#xxx	Immediate	4(1/0)	8(2/0)

*The size of the index register (ix) does not affect execution time

Table 24. MOVE BYTE AND WORD INSTRUCTION CLOCK PERIODS

Source	Destination								
	Dn	An	An@	An@ +	An@ −	An@(d)	An@(d,ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An@	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
An@ +	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
An@ −	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(3/1)	18(3/1)	22(4/1)
An@(d)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
An@(d, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx.W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx.L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
PC@(d)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
PC@(d, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#xxx	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

The size of the index register (ix) does not affect execution time

Table 25. MOVE LONG INSTRUCTION CLOCK PERIODS

Source	Destination								
	Dn	An	An@	An@ +	An@ −	An@(d)	An@(d,ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	14(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	14(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An@	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
An@ +	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
An@ −	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
An@(d)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
An@(d, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx.W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx.L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
PC@(d)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
PC@(d, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
#xxx	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)

*The size of the index register (ix) does not affect execution time.

Preliminary

Table 26. STANDARD INSTRUCTION CLOCK PERIODS

Instruction	Size	op <ea>, An	op <ea>, Dn	op Dn, <M>
ADD	Byte, Word	8(1/0) +	4(1/0) +	8(1/1) +
	Long	6(1/0) + **	6(1/0) + **	12(1/2) +
AND	Byte, Word	–	4(1/0) +	8(1/1) +
	Long	–	6(1/0) + **	12(1/2) +
CMP	Byte, Word	6(1/0) +	4(1/0) +	–
	Long	6(1/0) +	6(1/0) +	–
DIVS	–	–	158(1/0) + *	–
DIVU	–	–	140(1/0) + *	–
EOR	Byte, Word	–	4(1/0)***	8(1/1) +
	Long	–	8(1/0)***	12(1/2) +
MULS	–	–	70(1/0) + *	–
MULU	–	–	70(1/0) + *	–
OR	Byte, Word	–	4(1/0) +	8(1/1) +
	Long	–	8(1/0) + **	12(1/1) +
SUB	Byte, Word	8(1/0) +	4(1/0) +	8(1/1) +
	Long	6(1/0) + **	6(1/0) + **	12(1/2) +

+ add effective address calculation time

* indicates maximum value

** total of 8 clock periods for instruction if the effective address is register direct

*** only available effective address mode is data register direct

Table 27. IMMEDIATE INSTRUCTION CLOCK PERIODS

Instruction	Size	op #, Dn	op #, An	op #, M
ADDI	Byte, Word	8(2/0)	–	12(2/1) +
	Long	16(3/0)	–	20(3/2) +
ADDQ	Byte, Word	4(1/0)	8(1/0)*	8(1/1) +
	Long	8(1/0)	8(1/0)	12(1/2) +
ANDI	Byte, Word	8(2/0)	–	12(2/1) +
	Long	16(3/0)	–	20(3/1) +
CMPI	Byte, Word	8(2/0)	8(2/0)	8(2/0) +
	Long	14(3/0)	14(3/0)	12(3/0) +
EORI	Byte, Word	8(2/0)	–	12(2/1) +
	Long	16(3/0)	–	20(3/2) +
MOVEQ	Long	4(1/0)	–	–
ORI	Byte, Word	8(2/0)	–	12(2/1) +
	Long	16(3/0)	–	20(3/2) +
SUBI	Byte, Word	8(2/0)	–	12(2/1) +
	Long	16(3/0)	–	20(3/2) +
SUBQ	Byte, Word	4(1/0)	8(1/0)*	8(1/1) +
	Long	8(1/0)	8(1/0)	12(1/2) +

+ add effective address calculation time

*word only

Table 28. SINGLE OPERAND INSTRUCTION CLOCK PERIODS

Instruction	Size	Register	Memory
CLR	Byte, Word	4(1/0)	8(1/1) +
	Long	6(1/0)	12(1/2) +
NBCD	Byte	6(1/0)	8(1/1) +
NEG	Byte, Word	4(1/0)	8(1/1) +
	Long	6(1/0)	12(1/2) +
NEGX	Byte, Word	4(1/0)	8(1/1) +
	Long	6(1/0)	12(1/2) +
NOT	Byte, Word	4(1/0)	8(1/1) +
	Long	6(1/0)	12(1/2) +
Scc	Byte, False	4(1/0)	8(1/1) +
	Byte, True	6(1/0)	8(1/1) +
TAS	Byte	4(1/0)	10(1/1) +
TST	Byte, Word	4(1/0)	4(1/0)
	Long	4(1/0)	4(1/0) +

+ add effective address calculation time

Table 29. SHIFT AND ROTATE INSTRUCTION CLOCK PERIODS

Instruction	Size	Register	Memory
ASR, ASL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	—
LSR, LSL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	—
ROR, ROL	Byte, Word	6 - 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	—
ROXR, ROXL	Byte, Word	6 + 2n(1/0)	8(1/1) +
	Long	8 + 2n(1/0)	—

Table 30. BIT MANIPULATION INSTRUCTION CLOCK PERIODS

Instruction	Size	Dynamic		Static	
		Register	Memory	Register	Memory
BCHG	Byte	—	8(1/1) +	—	12(2/1) +
	Long	8(1/0) *	—	12(2/0) *	—
BCLR	Byte	—	8(1/1) +	—	12(2/1) +
	Long	10(1/0) *	—	14(2/0) *	—
BSET	Byte	—	8(1/1) +	—	12(2/1) +
	Long	8(1/0) *	—	12(2/0) *	—
BTST	Byte	—	4(1/0) +	—	8(2/0) +
	Long	6(1/0)	—	10(2/0)	—

+ add effective address calculation time

* indicates maximum value

Multi-Precision Instruction Clock Periods

Table 33 indicates the number of clock periods required for the multi-precision instructions. The number of clock periods include the time to fetch both operands, perform the operations, store the results, and read the next instructions. The number of bus read and write cycles is shown in parenthesis as (r/w).

In table 33, the headings have the following meaning: Dn = data register operand and M = memory operand.

Miscellaneous Instructions Clock Periods

Table 34 indicates the number of clock periods required for miscellaneous instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Exception Processing Clock Periods

Table 35 indicates the number of clock periods required for exception processing. The number of clock periods includes the time for all stacking, the vector fetch, and the fetch of the first instruction of the handler routine. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 31. CONDITIONAL INSTRUCTION CLOCK PERIODS

Instruction	Displacement	Trap or Branch Taken	Trap or Branch Not Taken
Bcc	Byte	10(2/0)	8(1/0)
	Word	10(2/0)	12(2/0)
BRA	Byte	10(2/0)	–
	Word	10(2/0)	–
BSR	Byte	18(2/2)	–
	Word	18(2/2)	–
DBcc	CC true	–	12(2/0)
	CC false	10(2/0)	14(3/0)
CHK	–	40(5/3) + *	8(1/0) +
TRAP	–	34(4/3)	–
TRAPV	–	34(5/3)	4(1/0)

+ add effective address calculation time

* indicates maximum value

Table 32. JMP, JSR, LEA, PEA, MOVEM INSTRUCTION CLOCK PERIODS

Instr	Size	An@	An@ -	An@ -	An@(d)	An@(d, ix) *	xxx.W	xxx.L	PC@(d)	PC@(d, ix)*
JMP	–	8(2/0)	–	–	10(2/0)	14(3/0)	10(2/0)	12(3/0)	10(2/0)	14(3/0)
JSR	–	16(2/2)	–	–	18(2/2)	22(2/2)	18(2/2)	20(3/2)	18(2/2)	22(2/2)
LEA	–	4(1/0)	–	–	8(2/0)	12(2/0)	8(2/0)	12(3/0)	8(2/0)	12(2/0)
PEA	–	12(1/2)	–	–	16(2/2)	20(2/2)	16(2/2)	20(3/2)	16(2/2)	20(2/2)
MOVEM M → R	Word	12 + 4n (3 + n/0)	12 + 4n (3 + n/0)	– –	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)	16 + 4n (4 + n/0)	20 + 4n (5 + n/0)	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)
	Long	12 + 8n (3 + 2n/0)	12 + 8n (3 + 2n/0)	– –	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)	16 + 8n (4 + 2n/0)	20 + 8n (5 + 2n/0)	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)
MOVEM R → M	Word	8 + 5n (2/n)	– –	8 + 5n (2/n)	12 + 5n (3/n)	14 + 5n (3/n)	12 + 5n (3/n)	16 + 5n (4/n)	–	– –
	Long	8 + 10n (2/2n)	– –	8 + 10n (2/2n)	12 + 10n (3/2n)	14 + 10n (3/2n)	12 + 10n (3/2n)	16 + 10n (4/2n)	–	– –

n is the number of registers to move

* the size of the index register (ix) does not affect the instruction's execution time

Table 33. MULTI-PRECISION INSTRUCTION CLOCK PERIODS

Instruction	Size	op Dn, Dn	op M, M
ADDX	Byte, Word	4(1/0)	18(3/1)
	Long	8(1/0)	30(5/2)
CMPM	Byte, Word	–	12(3/0)
	Long	–	20(5/0)
SUBX	Byte, Word	4(1/0)	18(3/1)
	Long	8(1/0)	30(5/2)
ABCD	Byte	6(1/0)	18(3/1)
SBCD	Byte	6(1/0)	18(3/1)

Table 34. MISCELLANEOUS INSTRUCTIONS CLOCK PERIODS

Instruction	Size	Register	Memory	Register → Memory	Memory → Register
MOVE from SR	–	6(1/0)	8(1/1)+	–	–
MOVE to CCR	–	12(2/0)	12(2/0)+	–	–
MOVE to SR	–	12(2/0)	12(2/0)+	–	–
MOVEP	Word	–	–	16(2/2)	16(4/0)
	Long	–	–	24(2/4)	24(6/0)
EXG	–	6(1/0)	–	–	–
EXT	Word	4(1/0)	–	–	–
	Long	4(1/0)	–	–	–
LINK	–	16(2/2)	–	–	–
MOVE from USP	–	4(1/0)	–	–	–
MOVE to USP	–	4(1/0)	–	–	–
NOP	–	4(1/0)	–	–	–
RESET	–	132(1/0)	–	–	–
RTE	–	20(5/0)	–	–	–
RTR	–	20(5/0)	–	–	–
RTS	–	16(4/0)	–	–	–
STOP	–	4(0/0)	–	–	–
SWAP	–	4(1/0)	–	–	–
UNLK	–	12(3/0)	–	–	–

+ add effective address calculation time

Table 35. EXCEPTION PROCESSING CLOCK PERIODS

Exception	Periods
Address Error	50(4/7)
Bus Error	50(4/7)
Interrupt	44(5/3)*
Illegal Instruction	34(4/3)
Privileged Instruction	34(4/3)
Trace	34(4/3)

* The interrupt acknowledge bus cycle is assumed to take four external clock periods

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltage	− 0.3 to + 7.0	V
Input voltage[3]	− 0.3 to + 7.0	V
Operating temperature range[2]	0 to + 70	°C
Storage temperature	− 55 to + 150	°C

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V ± 5%, V_{SS} = 0V; T_A = 0°C to + 70°C (see figures 37-39)[4,5]

	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IH}	Input high voltage		2.0	V_{CC}	V
V_{IL}	Input low voltage		V_{SS}-0.75	0.8	V
I_{in}	Input leakage current BERR, BGACK, BR, DTACK, CLK IPO0-IPL2, VPA HALT, RESET	5.25V		 2.5 20	 μA μA
I_{TSI}	Three-state (off state) input current AS, A1-A23, D0-D15, FC0-FC2, LDS, R/W, UDS, VMA	2.4V/0.4V		 20	 μA
V_{OH}	Output high voltage E[6] AS, A1-A23, BG, D0-D15, FC0-FC2, LDS, R/W, UDS, VMA	I_{OH} = − 400μA	 V_{CC}-0.75 2.4		 V V
V_{OL}	Output low voltage HALT A1-A23, BG, FC0-FC2 RESET E, AS, D0-D15, LDS, R/W, UDS, VMA	 I_{OL} = 1.6mA I_{OL} = 3.2mA I_{OL} = 35.0mA I_{OL} = 5.3mA		 0.5 0.5 0.5 0.5	 V V V V
P_D	Power dissipation	Clock frequency = *MHz		1.5	W
C_{in}	Capacitance	V_{in} = 0V, T_A = 25°C, frequency = 1MHz		10.0	pF

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on + 150°C maximum junction temperature and thermal resistance of 30°C/W junction to ambient for ceramic package.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. With external pullup resistor of 470 ohms.
7. For a loading capacitance of less than or equal to 500pF, subtract 5ns from the values given in these columns.
8. Actual value depends on clock period.
9. If #47 is satisfied for both DTACK and BERR, #48 can be 0ns.
10. After V_{CC} has been applied for 100ms.
11. If the asynchronous setup time (#47) requirements are satisfied, the DTACK low-to-data setup time (#31) requirements can be ignored. The data must only satisfy the data-in to clock low setup time (#27) for the following cycle.

16-BIT MICROPROCESSOR

SC68000 SERIES

Preliminary

AC ELECTRICAL SPECIFICATIONS $V_A = 5VDC$, $T_A = 0°C$ to $70°C$ (see figures 41-45)[4,5]

NUMBER	CHARACTERISTIC	SYMBOL	TENTATIVE LIMITS								UNIT
			4MHz		6MHz		8MHz		10MHz		
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Clock period	t_{cyc}	250	500	167	500	125	500	100	500	ns
2	Clock width low	t_{CL}	115	250	75	250	55	250	45	250	ns
3	Clock width high	t_{CH}	115	250	75	250	55	250	45	250	ns
4	Clock fall time	t_{Cf}		10		10		10		10	ns
5	Clock rise time	t_{Cr}		10		10		10		10	ns
6	Clock low to address	t_{CLAV}		90		80		70		55	ns
6A	Clock high to FC valid	t_{CHFCV}		90		80		70		60	ns
7	Clock high to address data high impedance (maximum)	t_{CHAZx}		120		100		80		70	ns
8	Clock high to address/FC invalid (minimum)	t_{CHAZn}	0		0		0		0		ns
9[7]	Clock high to AS, DS low (maximum)	t_{CHSLx}		80		70		60		55	ns
10	Clock high to AS, DS low (minimum)	t_{CHSLn}	0		0		0		0		ns
11[8]	Address to AS, DS (read) low/AS write	t_{AVSL}	55		35		30		20		ns
11A[8]	FC valid to AS, DS (read) low/AS write	t_{FCVSL}	80		70		60		50		ns
12[7]	Clock low to AS, DS high	t_{CLSH}		90		80		70		55	ns
13[8]	AS, DS high to address/FC invalid	t_{SHAZ}	60		40		30		20		ns
14[8]	AS, DS width low (read)/AS write	t_{SL}	535		337		240		195		ns
14A[8]	DS width low (write)		285		170		115		95		ns
15[7]	AS, DS width high	t_{SH}	285		180		150		105		ns
16	Clock high to AS, DS high impedance	t_{CHSZ}		120		100		80		70	ns
17[8]	AS, DS high to R/W high	t_{SHRH}	60		50		40		20		ns
18[7]	Clock high to R/W high (maximum)	t_{CHRHx}		90		80		70		60	ns
19	Clock high to R/W high (minimum)	t_{CHRHn}	0		0		0		0		ns
20[7]	Clock high to R/W low	t_{CHRL}		90		80		70		60	ns
21[8]	Address valid to R/W low	t_{AVRL}	45		25		20		0		ns
21A[8]	FC valid to R/W low	t_{FCVRL}	80		70		60		50		ns
22[8]	R/W low to DS low (write)	t_{RLSL}	200		140		80		50		ns
23	Clock low to data out valid	t_{CLDO}		90		80		70		55	ns
24	Clock high to R/W, VMA high impedance	t_{CHRZ}		120		100		80		70	ns
25[8]	DS high to data out invalid	t_{SHDO}	60		40		30		20		ns
26[8]	Data out valid to DS low (write)	t_{DOSL}	55		35		30		20		ns
27[11]	Data in to clock low (setup time)	t_{DICL}	30		25		15		15		ns
28[8]	AS, DS high to DTACK high	t_{SHDAH}	0	240	0	160	0	120	0	90	ns
29	DS high to data invalid (hold time)	t_{SHDI}	0		0		0		0		ns
30	AS, DS high to BERR high	t_{SHBEH}	0		0		0		0		ns
31[8]	DTACK low to data in (setup time)	t_{DALDI}		180		120		90		65	ns
32	HALT and RESET input transition time	t_{RHrf}	0	200	0	200	0	200	0	200	ns
33	Clock high to BG low	t_{CHGL}		90		80		70		60	ns
34	Clock high to BG high	t_{CHGH}		90		80		70		60	ns
35	BR low to BG low	t_{BRLGL}	1.5	3.0	1.5	3.0	1.5	3.0	1.5	3.0	Clk Per
35	BR low to BG low (figure 43)	t_{BRLGL}	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per

Preliminary

AC ELECTRICAL SPECIFICATIONS (Continued) $V_A = 5VDC$, $T_A = 0°C$ to $70°C$ (see figures 41-45)[4,5]

NUMBER	CHARACTERISTIC	SYMBOL	TENTATIVE LIMITS								UNIT
			4MHz		6MHz		8MHz		10MHz		
			Min	Max	Min	Max	Min	Max	Min	Max	
36	BR high to BG high	t_{BRHGH}	1.5	3.0	1.5	3.0	1.5	3.0	1.5	3.0	Clk Pe
37	BGACK low to BG high	t_{GALGH}	1.5	3.0	1.5	3.0	1.5	3.0	1.5	3.0	Clk Per
38	BG low to bus high impedance (AS high)	t_{GLZ}		120		100		80		70	ns
39	BG width high	t_{GH}	1.5		1.5		1.5		1.5		Clk Per
40	Clock low to VMA low	t_{CLVML}		90		80		70		70	ns
41	Clock low to E transition	t_{CLC}		100		85		70		55	ns
42	E output rise and fall time	t_{Erf}		25		25		25		25	ns
43	VMA low to E high	t_{VMLEH}	325		240		200		150		ns
44	AS, DS high to VPA high	t_{SHVPH}	0	240	0	160	0	120	0	90	ns
45	E low to address/VMA/FC invalid	t_{ELAI}	55		35		30		10		ns
46	BGACK width	t_{BGL}	1.5		1.5		1.5		1.5		Clk Per
47[11]	Asynchronous input setup time	t_{ASI}	30		25		20		20		ns
48[9]	BERR low to DTACK low	t_{BELDAL}	50		50		50		50		ns
49	E low to AS, DS invalid	t_{ELSI}	-80		-80		-80		-80		ns
50	E width high	t_{EH}	900		600		450		350		ns
51	E width low	t_{EL}	1400		900		700		550		ns
52	E extended rise time	t_{CIEHX}	80		80		80		80		ns
53	Data hold from clock high	t_{CHDO}	0		0		0		0		ns
54	Data hold from E low (write)	t_{ELDOZ}	60		40		30		20		ns
55	R/W to data bus impedance change	t_{RLDO}	55		35		30		20		ns
56[10]	Halt/RESET pulse width	t_{HRPW}	10		10		10		10		Clk Pe

CLOCK TIMING (see figure 40)

CHARACTERISTIC	SYMBOL	4MHz		6MHz		8MHz		10MHz		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
Frequency of Operation	F	2.0	4.0	2.0	6.0	2.0	8.0	2.0	10.0	MHz
Cycle Time	t_{cyc}	250	500	167	500	125	500	100	500	ns
Clock Pulse Width	t_{CL} t_{CH}	115 115	250 250	75 75	250 250	55 55	250 250	45 45	250 250	ns
Rise and Fall Times	t_{Cr} t_{Cf}	— —	10 10	— —	10 10	— —	10 10	— —	10 10	ns

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$$P_D = P_{INT} + P_{I/O} \quad (2)$$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

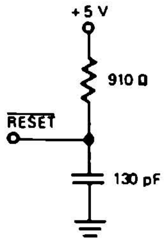

+5 V
910 Ω
$\overline{RESET}$
130 pF

Figure 37. Reset Test Load

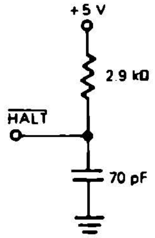
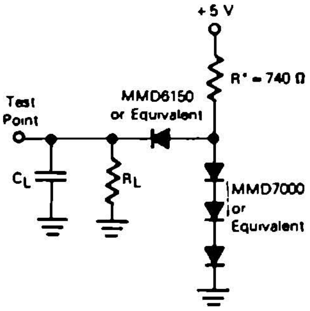

+5 V
2.9 kΩ
$\overline{HALT}$
70 pF

Figure 38. Halt Test Load


+5 V
R* = 740 Ω
Test Point
MMD6150 or Equivalent
C_L
R_L
MMD7000 or Equivalent

C_L = 130 pF
(Includes all Parasitics)
R_L = 6.0 kΩ for
$\overline{AS}$, A1-A23, $\overline{BG}$, D0-D15, E
FC0-FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$
*R = 1.22 kΩ for A1-A23, $\overline{BG}$,
E, FC0-FC2

Figure 39. Test Loads

All timing diagrams should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

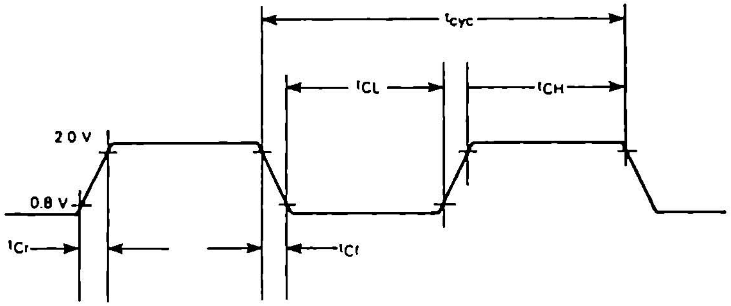
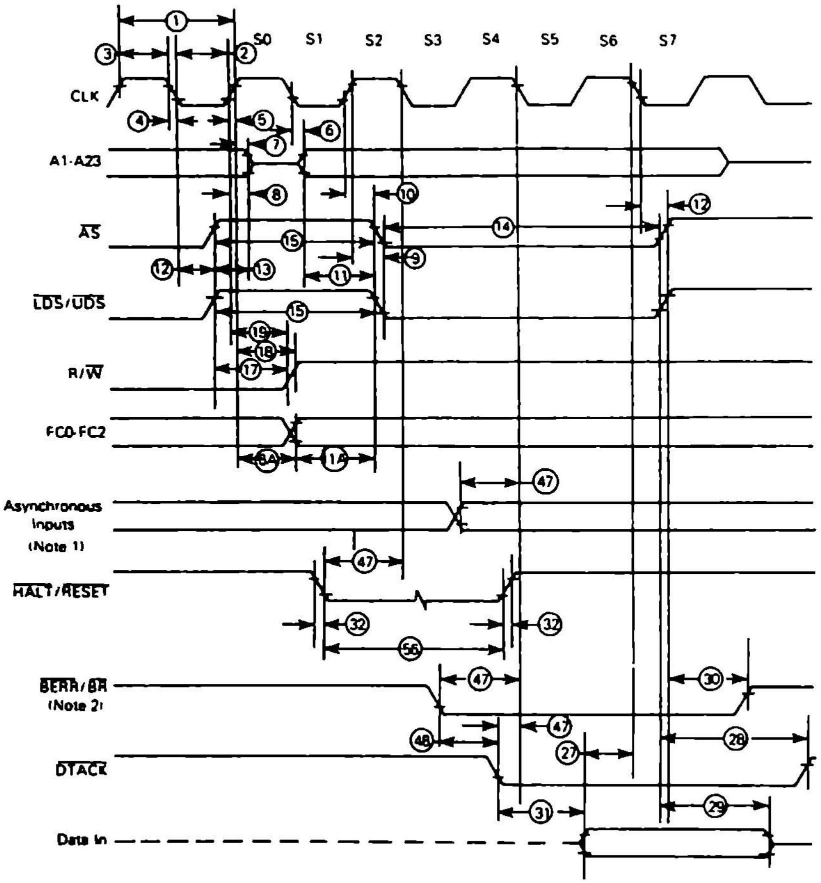

t_{cyc}
t_{CL}
t_{CH}
2.0 V
0.8 V
t_{Cr}
t_{Cf}

Figure 40. Input Clock Timing


S0
S1
S2
S3
S4
S5
S6
S7
CLK
A1-A23
$\overline{AS}$
$\overline{LDS}/\overline{UDS}$
R/$\overline{W}$
FC0-FC2
Asynchronous Inputs (Note 1)
$\overline{HALT}/\overline{RESET}$
$\overline{BERR}/\overline{BR}$ (Note 2)
$\overline{DTACK}$
Data In
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
⑭
⑮
⑰
⑱
⑲
6A
1A
(47)
(32)
(56)
(48)
㉗
㉘
㉙
㉚
㉛

NOTES:

1 Setup time for the asynchronous inputs $\overline{BGACK}$, $\overline{IPL0}$-$\overline{IPL2}$, and $\overline{VPA}$ guarantees their recognition at the next falling edge of the clock

2 $\overline{BR}$ need fall at this time only in order to insure being recognized at the end of this bus cycle

Figure 41. Read Cycle Timing

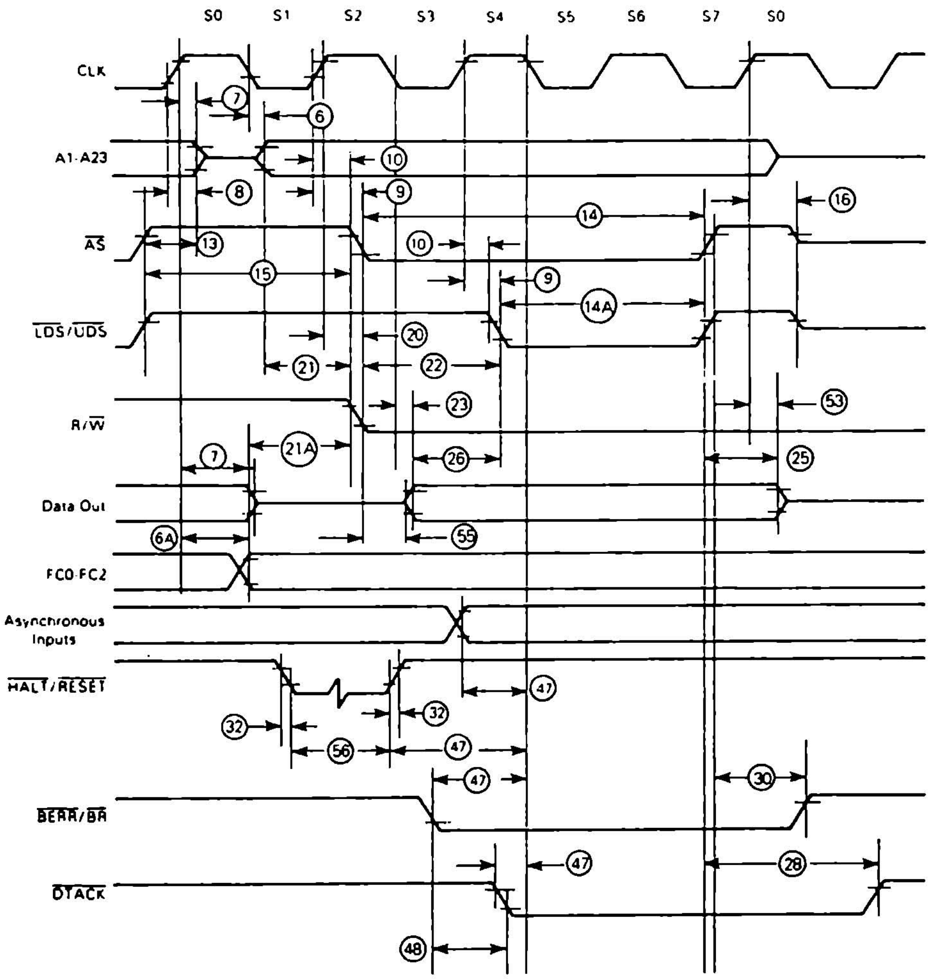
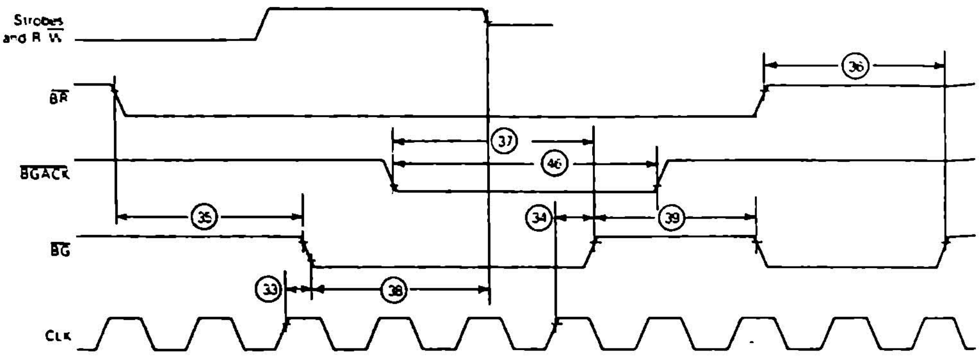

S0
S1
S2
S3
S4
S5
S6
S7
S0
CLK
A1-A23
AS
LDS/UDS
R/W
Data Out
FC0-FC2
Asynchronous Inputs
HALT/RESET
BERR/BR
DTACK
7
6
10
8
9
16
14
13
10
15
9
14A
20
21
22
23
53
7
21A
26
25
55
6A
47
32
32
56
47
47
30
47
28
48

Figure 42. Write Cycle Timing

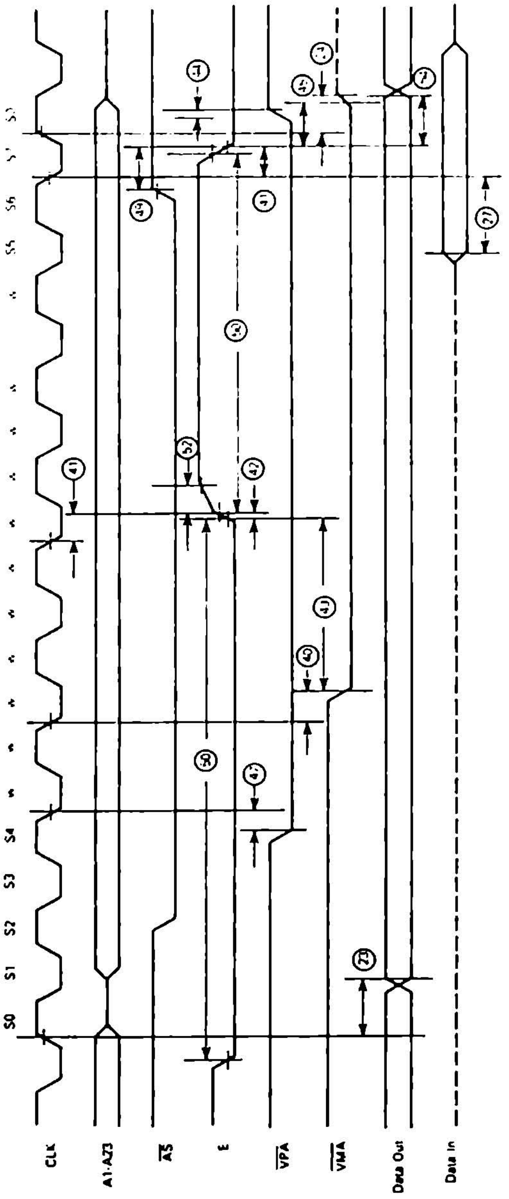
Preliminary


Strobes and R/W
BR
BGACK
BG
CLK
36
37
46
35
34
39
33
38

NOTES

1 Setup time for the asynchronous inputs BERR, BGACK, BR, DTACK, IPL0-IPL2, and VPA guarantees their recognition at the next falling edge of the clock

Figure 43. AC Electrical Timing—Bus Arbitration


CLK
A1-A23
$\overline{AS}$
E
$\overline{VPA}$
$\overline{VMA}$
Data Out
Data In
S0
S1
S2
S3
S4
S5
S6
S7
23
47
50
49
48
41
42
52
27

NOTE: This figure represents the best case synchronous timing where $\overline{VPA}$ falls before the third system clock cycle after the falling edge of E

Figure 44. Synchronous Timing—Best Case

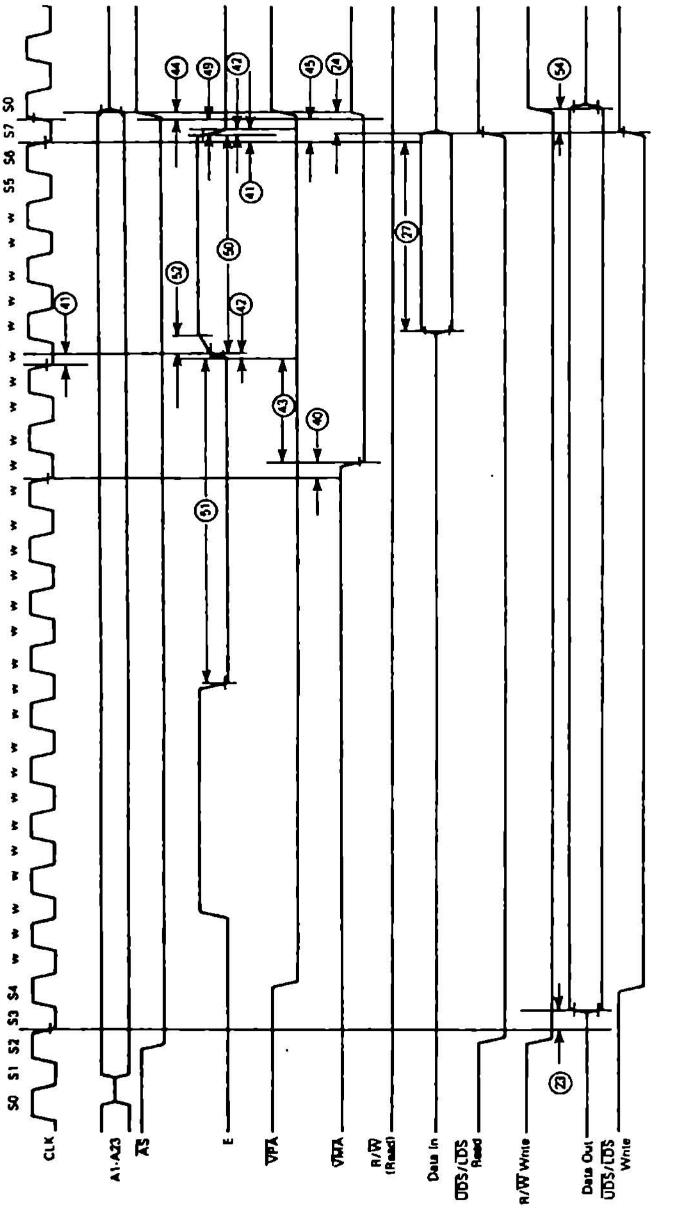

CLK
A1-A23
$\overline{AS}$
E
$\overline{VPA}$
$\overline{VMA}$
R/$\overline{W}$ (Read)
Data In
$\overline{UDS}$/$\overline{LDS}$ Read
R/$\overline{W}$ Write
Data Out
$\overline{UDS}$/$\overline{LDS}$ Write
S0 S1 S2 S3 S4 w w w w w w w w w w w w w w w w w w w w w w w w w S5 S6 S7 S0
44
49
42
45
24
54
41
27
50
52
41
42
43
40
51
23

Figure 45. Synchronous Timing—Worst Case

FEBRUARY 1982

PARALLEL INTERFACE/TIMER

SC68230

Preliminary

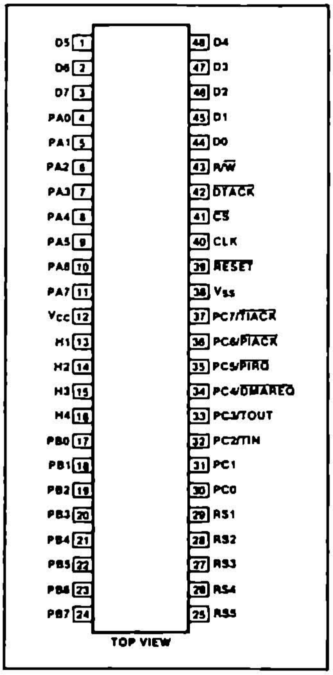
DESCRIPTION

The SC68230 Parallel Interface/Timer (PI/T) provides versatile double buffered parallel interfaces and an operating system oriented timer to S68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes, either 8 or 16 bits wide. In the unidirectional modes, an associated data direction register determines whether the port pins are inputs or outputs. In the bidirectional modes, the data direction registers are ignored and the direction is determined dynamically by the state of four handshake pins. These programmable handshake pins provide an interface flexible enough for connection to a wide variety of low, medium, or high speed peripherals or other computer systems. The PI/T ports allow use of vectored or autovectored interrupts, and also provide a DMA request pin for connection to direct memory access controllers. The PI/T timer contains a 24-bit counter and a 5-bit prescaler. The timer can be clocked by the system clock (PI/T CLK pin) or by an external clock (TIN pin), and a 5-bit prescaler can be used. It can generate periodic interrupts, a square wave or a single interrupt after a programmed time period. Also it can be used for elapsed time measurement or as a device watchdog. Table 1 is a summary of the input and output signals and Figure 1 shows the functional pin assignments. Figure 2 is a PI/T system block diagram.

FEATURES

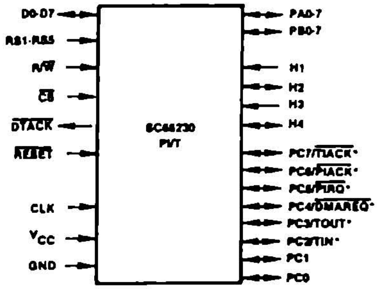
- **S68000 bus compatible**
- **Port modes include:**
  Bit I/O
  Unidirectional 8-bit and 16-bit
  Bidirectional 8-bit and 16-bit
- **Selectable handshaking options**
- **24-Bit programmable timer**
- **Software programmable timer modes**
- **Contains interrupt vector generation logic**
- **Separate port and timer interrupt service requests**
- **Registers are read/write and directly addressable**
- **Registers are addressed for MOVEP (move peripheral) and DMAC compatibility**

PIN CONFIGURATION[1]


D5 1
D6 2
D7 3
PA0 4
PA1 5
PA2 6
PA3 7
PA4 8
PA5 9
PA6 10
PA7 11
V_{CC} 12
H1 13
H2 14
H3 15
H4 16
PB0 17
PB1 18
PB2 19
PB3 20
PB4 21
PB5 22
PB6 23
PB7 24
48 D4
47 D3
46 D2
45 D1
44 D0
43 R/$\overline{W}$
42 $\overline{DTACK}$
41 $\overline{CS}$
40 CLK
39 $\overline{RESET}$
38 V_{SS}
37 PC7/$\overline{TIACK}$
36 PC6/$\overline{PIACK}$
35 PC5/$\overline{PIRQ}$
34 PC4/$\overline{DMAREQ}$
33 PC3/TOUT
32 PC2/TIN
31 PC1
30 PC0
29 RS1
28 RS2
27 RS3
26 RS4
25 RS5
TOP VIEW

ORDERING CODE

Packages	V_{CC} = 5V ± 5%, T_A = 0° to 70°C	
	8 MHz	10 MHz
Ceramic DIP	SC68230L8I48	SC68230LAI48
Plastic DIP	SC68230L8N48	SC68230LAN48


D0-D7
RS1-RS5
R/$\overline{W}$
$\overline{CS}$
$\overline{DTACK}$
$\overline{RESET}$
CLK
V_{CC}
GND
SC68230
PI/T
PA0-7
PB0-7
H1
H2
H3
H4
PC7/$\overline{TIACK}$*
PC6/$\overline{PIACK}$*
PC5/$\overline{PIRQ}$*
PC4/$\overline{DMAREQ}$*
PC3/TOUT*
PC2/TIN*
PC1
PC0

*Individually Programmable Dual-Function Pin.

Figure 1. Functional Pin Assignment

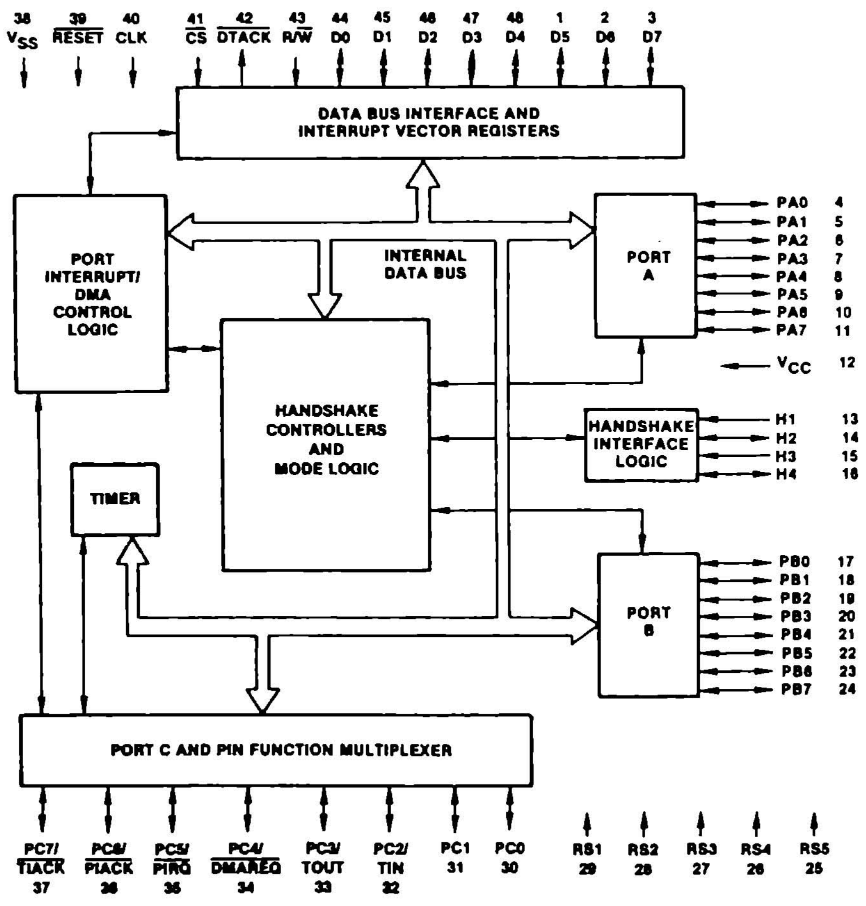
---

[1] In this data sheet, barring signal names (overscore) to indicate low is done only for the pin configuration diagram, signal description headings, tables and figures.

PARALLEL INTERFACE/TIMER

SC68230

Preliminary


38 V_{SS}
39 $\overline{RESET}$
40 CLK
41 $\overline{CS}$
42 $\overline{DTACK}$
43 R/$\overline{W}$
44 D0
45 D1
46 D2
47 D3
48 D4
1 D5
2 D6
3 D7
DATA BUS INTERFACE AND INTERRUPT VECTOR REGISTERS
PORT INTERRUPT/ DMA CONTROL LOGIC
INTERNAL DATA BUS
PORT A
PA0 4
PA1 5
PA2 6
PA3 7
PA4 8
PA5 9
PA6 10
PA7 11
V_{CC} 12
HANDSHAKE CONTROLLERS AND MODE LOGIC
HANDSHAKE INTERFACE LOGIC
H1 13
H2 14
H3 15
H4 16
TIMER
PORT B
PB0 17
PB1 18
PB2 19
PB3 20
PB4 21
PB5 22
PB6 23
PB7 24
PORT C AND PIN FUNCTION MULTIPLEXER
PC7/$\overline{TIACK}$ 37
PC6/$\overline{PIACK}$ 36
PC5/$\overline{PIRQ}$ 35
PC4/$\overline{DMAREQ}$ 34
PC3/TOUT 33
PC2/TIN 32
PC1 31
PC0 30
RS1 29
RS2 28
RS3 27
RS4 26
RS5 25

PARALLEL INTERFACE/TIMER SC68230

Preliminary

Table 1 SIGNAL SUMMARY

Signal Name	Mnemonic	Input/Output	Active State	Three State
Bidirectional Data Bus	D0-D7	input/output	high	yes
Register Selects	RS1-RS5	input	high	—
Read/Write Input	R/W	input	read-high write-low	—
Chip Select	$\overline{CS}$	input	low	—
Data Transfer Acknowledge	$\overline{DTACK}$	output	low	yes
Reset	$\overline{RESET}$	input	low	—
Clock Input	CLK	input	high	—
Port A and Port B	PA0-PA7, PB0-PB7	input/output	high	yes
Handshake	H1, H3	input	programmable	—
Handshake	H2, H4	input/output	programmable	yes
Port C	PC0-PC7	input/output	high	yes
Ground	V_{SS}	input	—	—
Power Input	V_{CC}	input	—	—

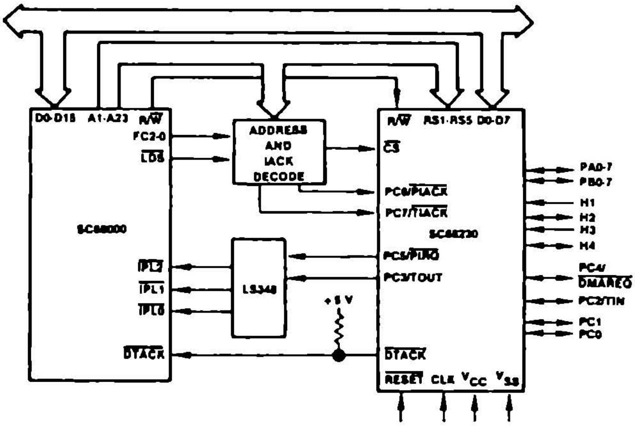

D0-D15
A1-A23
R/$\overline{W}$
FC2-0
$\overline{LDS}$
SC68000
$\overline{IPL2}$
$\overline{IPL1}$
$\overline{IPL0}$
$\overline{DTACK}$
ADDRESS AND IACK DECODE
LS348
+5 V
R/$\overline{W}$
RS1-RS5
D0-D7
$\overline{CS}$
PC6/$\overline{PIACK}$
PC7/$\overline{TIACK}$
SC68230
PC5/$\overline{PIRQ}$
PC3/TOUT
$\overline{DTACK}$
$\overline{RESET}$
CLK
V_{CC}
V_{SS}
PA0-7
PB0-7
H1
H2
H3
H4
PC4/$\overline{DMAREQ}$
PC2/TIN
PC1
PC0

Figure 2. PI/T System Block Diagram

FEBRUARY [illegible]

PARALLEL INTERFACE/TIMER

SC68230

Preliminary

GENERAL DESCRIPTION

The PI/T consists of two logically independent sections; the ports and the timer. The port section consists of port A (PA0-7), port B (PB0-7), four handshake pins (H1, H2, H3, and H4), two general I/0 pins, and six dual-function pins. The dual-function pins can individually operate as a third port (port C) or an alternate function related to either ports A and B, or the timer. The four programmable handshake pins, depending on the mode, can control data transfer to and from the ports, or can be used as interrupt generating inputs, or I/0 pins.

The timer consists of a 24-bit counter, optionally clocked by a 5-bit prescaler. Three pins provide complete timer I/0: PC2/TIN, PC3/TOUT, and PC7/TIACK. In specific applications, only the pins needed for the given configuration perform the timer function, while the others remain port C I/0.

The system bus interface provides for asynchronous transfer of data from the PI/T to a bus master over the data bus (D0-D7). Data transfer acknowledge (DTACK), register selects (RS1-RS5), chip select, the read/write line (R/W), and port interrupt acknowledge (PIACK) or timer interrupt acknowledge (TIACK) control data transfer between the PI/T and the SC68000.

PIN DESCRIPTIONS

D0-D7
Bidirectional data bus. The data bus pins D0-D7 form an 8-bit bidirectional data bus to/from the SC68000 or other bus master. These pins are active high.

RS1-RS5
Register selects. RS1-RS5 are active high, high-impedance inputs that determine which of the 25 internal PI/T registers is being addressed.

R/W
Read/write input. R/W is the read/write signal from the SC68000 or bus master, indicating whether the current bus cycle is a read (high) or write (low) cycle.

CS
Chip select input. The CS input selects the PI/T registers for the current bus cycle. Address strobe and the data strobe (upper or lower) of the bus master, along with the appropriate address bits, must be included in the chip select equation. A low level corresponds to an asserted chip select.

DTACK
Data transfer acknowledge output. DTACK is an active low output that signals the completion of the bus cycle. During read or interrupt acknowledge cycles, DTACK is asserted by the SC68230 after data has been provided on the data bus; during write cycles it is asserted after data has been accepted at the data bus. Data transfer acknowledge is compatible with the SC68000 and with other bus masters such as the SC68430 DMA controller. A holding resistor is required to maintain DTACK high between bus cycles.

RESET
Reset input. RESET is used to initialize all PI/T functions. All control and data direction registers are cleared and most internal operations are disabled by the assertion of RESET (low).

CLK
Clock input. The clock pin is a TTL-compatible input with the same specifications as the SC68000. The PI/T contains dynamic logic throughout, and hence this clock must not be gated off at any time. It is not necessary that this clock maintain any particular phase relationship with the SC68000 clock. It can be connected to an independent frequency source (faster or slower) as long as all bus specifications are met.

PA0-PA7 and PB0-PB7
Port A and port B. Ports A and B are 8-bit ports that can be concatenated to form a 16-bit port in certain modes. The ports can be controlled in conjunction with the handshake pins H1-H4. For stabilization during system power-up, ports A and B have internal pullup resistors to V_{CC}. All port pins are active high.

H1-H4
Handshake pins (I/0 depending on the mode and submode). Handshake pins H1-H4 are multi-purpose pins that (depending on the operational mode) can provide an interlocked handshake, a pulsed handshake, an interrupt input (independent of data transfers), or simple I/0 pins. For stabilization during system power-up, H2 and H4 have internal pullup resistors to V_{CC}. Their sense (active high or low) can be programmed in the port general control register bits 3-0. Independent of the mode, the instantaneous level of the handshake pins can be read from the port status register.

Port C
(PC0-PC7/alternate function). This port can be used as eight general purpose I/0 pins (PC0-PC7) or any combination of six special function pins and two general purpose I/0 pins (PC0-PC1). (Each dual function pin can be standard I/0 or a special function independent of the other port C pins.) When used as a port C pin, these pins are active high. They can be individually programmed as inputs or outputs by the Port C data direction register.

The alternate functions (TIN, TOUT, and TIACK) are timer I/0 pins. TIN can be used as a rising-edge triggered external clock input or an external run/halt control pin (the timer is in the run state if run/halt is high and in the halt state if run/halt is low). TOUT can provide an active low timer interrupt request output or a general-purpose square-wave output, initially high. TIACK is an active low input used for timer interrupt acknowledge.

Port A and B functions have an independent pair of active low interrupt request (PIRQ) and interrupt acknowledge (PIACK) pins.

The DMAREQ (direct memory access request) pin provides an active low direct memory access controller (DMAC) request pulse of three clock cycles.

REGISTER MODEL

A register model that includes the corresponding register selects is shown in Table 2.

PORT FUNCTIONAL DESCRIPTION

Port Control Structure
The primary focus of most applications will be on ports A and B, the handshake pins, the port interrupt pins, and the DMA request pin. The ports are controlled by the port general control register which contains a 2-bit field that specifies a set of four operation modes. These govern the overall operation of the ports and determine their interrelationships. Some modes require additional information from each port's control register to further define its operation. In each port control register, there is a 2-bit submode field that serves this purpose. Each port mode/submode combination specifies a set of programmable characteristics that fully define the behavior of that port and two of the handshake pins. This structure is summarized in Table 3 and Figure 3.

Preliminary

Table 2 REGISTER MODEL

Register Select Bits 5	4	3	2	1	7	6	5	4	3	2	1	0	
0	0	0	0	0	Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	Port General Control Register
0	0	0	0	1	*	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control			Port Service Request Register
0	0	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Data Direction Register
0	0	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Data Direction Register
0	0	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port C Data Direction Register
0	0	1	0	1	Interrupt Vector Number						*	*	Port Interrupt Vector Register
0	0	1	1	0	Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl	Port A Control Register
0	0	1	1	1	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl	Port B Control Register
0	1	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Data Register
0	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Data Register
0	1	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Alternate Register
0	1	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Alternate Register
0	1	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port C Data Register
0	1	1	0	1	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S	Port Status Register
0	1	1	1	0	*	*	*	*	*	*	*	*	(null)
0	1	1	1	1	*	*	*	*	*	*	*	*	(null)
1	0	0	0	0	TOUT/$\overline{\text{TIACK}}$ Control			Z D Ctrl	*	Clock Control		Timer Enable	Timer Control Register
1	0	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Timer Interrupt Vector Register
1	0	0	1	0	*	*	*	*	*	*	*	*	(null)
1	0	0	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Counter Preload Register (High)
1	0	1	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	(Mid)
1	0	1	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Low)
1	0	1	1	0	*	*	*	*	*	*	*	*	(null)
1	0	1	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Count Register (High)
1	1	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	(Mid)
1	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Low)
1	1	0	1	0	*	*	*	*	*	*	*	ZDS	Timer Status Register
1	1	0	1	1	*	*	*	*	*	*	*	*	(null)
1	1	1	0	0	*	*	*	*	*	*	*	*	(null)
1	1	1	0	1	*	*	*	*	*	*	*	*	(null)
1	1	1	1	0	*	*	*	*	*	*	*	*	(null)
1	1	1	1	1	*	*	*	*	*	*	*	*	(null)

*Unused, read as zero.

Table 3 PORT MODE CONTROL SUMMARY

- Mode 0 (Unidirectional 8 Bit Mode)
  - Port A
    - Submode 00 – Double-Buffered Input
      - H1 – Latches input data
      - H2 Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed input handshake protocols
    - Submode 01 – Double Buffered Output
      - H1 Indicates data received by peripheral
      - H2 – Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed output handshake protocols
    - Submode 1X – Bit I/O
      - H1 – Status/interrupt generating input
      - H2 – Status/interrupt generating input or general purpose output
  - Port B, H3 and H4 – identical to Port A, H1 and H2
- Mode 1 (Unidirectional 16-Bit Mode)
  - Port A – Double Buffered Data (Most significant)
    - Submode XX (not used)
      - H1 – Status/interrupt generating input
      - H2 Status/interrupt generating input or general purpose output
  - Port B – Double Buffered Data (Least significant)
    - Submode X0 – Unidirectional 16 Bit Input
      - H3 – Latches input data
      - H4 – Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed input handshake protocols
    - Submode X1 – Unidirectional 16-Bit Output
      - H3 – Indicates data received by peripheral
      - H4 – Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed output handshake protocols
- Mode 2 (Bidirectional 8 Bit Mode)
  - Port A – Bit I/O (with no handshaking pins)
    - Submode XX (not used)
  - Port B – Bidirectional 8 Bit Data (Double-Buffered)
    - Submode XX (not used)
      - H1 – Indicates output data received by peripheral
      - H2 – Operation with H1 in the interlocked or pulsed output handshake protocols
      - H3 – Latches input data
      - H4 – Operation with H3 in the interlocked or pulsed input handshake protocols
- Mode 3 (Bidirectional 16 Bit Mode)
  - Port A – Double-Buffered Data (Most significant)
    - Submode XX (not used)
  - Port B – Double Buffered Data (Least significant)
    - Submode XX (not used)
  - H1 Indicates output data received by peripheral
  - H2 – Operation with H1 in the interlocked or pulsed output handshake protocols
  - H3 – Latches input data
  - H4 Operation with H3 in the interlocked or pulsed input handshake protocols

Port General Information and Conventions The following paragraphs introduce concepts that are generally applicable to the PI/T ports independent of the chosen mode and submode. For this reason no particular port or handshake pins are mentioned; the notation H1 (H3) indicates that depending on the chosen mode and submode, the statement given may be true for either the H1 or H3 handshake pin.

Unidirectional vs Bidirectional—Figure 3 shows the configuration of ports A and B and each of the handshake pins in each port mode and submode. In modes 0 and 1, a data direction register is associated with each of the ports. These registers contain one bit for each port pin to determine whether that pin is an input or an output. Modes 0 and 1 are, thus, called unidirectional modes because each pin assumes a constant direction, changeable only by a reset condition or a programming change. These modes allow double buffered data transfers in one direction. This direction, determined by the mode and submode definition, is known as the primary direction. Data transfers in the primary direction are controlled by the handshake pins. Data transfers not in the primary direction are generally unrelated and single or unbuffered data paths exist.

In modes 2 and 3 there is no concept of primary direction as in modes 0 and 1. Except for port A in mode 2 (bit I/O), the data direction registers have no effect. These modes are bidirectional, in that the direction of each transfer (always 8 or 16 bits double buffered) is determined dynamically by the state of the handshake pins. Thus, for example, data may be transferred out of the ports, followed very shortly by a transfer into the same port pins. Transfers to and from the ports are independent and may occur in any sequence. Since the instantaneous direction is always determined by the external system, a small amount of arbitration logic may be required.

Control of Double Buffered Data Paths—Generally speaking, the PI/T is a double buffered device. In the primary direction double buffering allows orderly transfers by using the handshake pins in any of several programmable protocols. (When bit I/O is used, double buffering is not available and the handshake pins are used as outputs or status/interrupt inputs.)

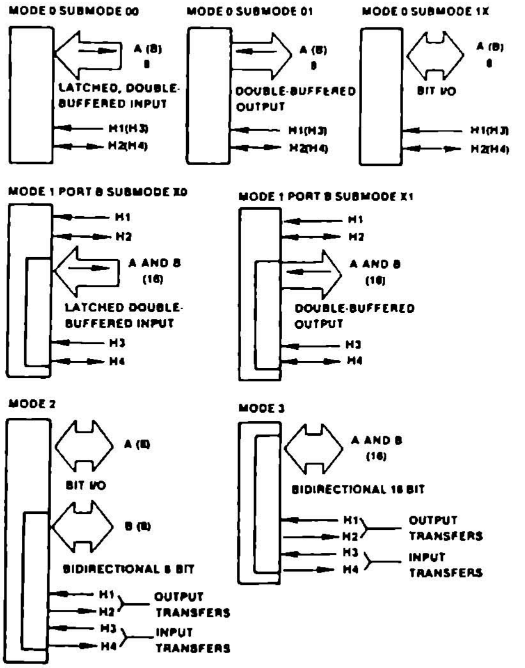


Figure 3. Port Mode Layout

Use of double buffering is most beneficial in situations where a peripheral device and the computer system are capable of transferring data at roughly the same speed. Double buffering allows the fetch operation of the data transmitter to be overlapped with the store operation of the data receiver. Thus, throughput measured in bytes or words per second can be greatly enhanced. If there is a large mismatch in transfer capability between the computer and the peripheral, little or no benefit is obtained. In these cases there is no penalty in using double buffering.

Double Buffered Input Transfers—In all modes, the PI/T supports double buffered input transfers. Data that meets the port setup and hold times is latched on the asserted edge of H1(H3). H1(H3) is edge sensitive, and may assume any duty cycle as long as both high and low minimum times are observed. The PI/T contains a port status register whose H1S(H3S) status bit is set anytime any input data is present in the double buffered latches that has not been read by the bus master. The action of H2(H4) is programmable; it may indicate whether there is room for more data in the PI/T latches or it may serve other purposes. The following options are available, depending on the mode.

1. H2(H4) may be an edge sensitive input that is independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by the direct method (refer to Direct Method of Resetting Status), the RESET pin being asserted, or when the H12 enable (H34 enable) bit of the port general control register is 0.
2. H2(H4) may be a general purpose output pin that is always negated. The H2S(H4S) status bit is always 0.
3. H2(H4) may be a general purpose output pin that is always asserted. The H2S(H4S) status bit is always 0.
4. H2(H4) may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H1(H3) input. As soon as the input latches become ready, H2(H4) is again asserted. When the input double buffered latches are full, H2(H4) remains negated until data is removed. Thus, anytime the H2(H4) output is asserted, new input data may be entered by asserting H1(H3). At other times, transitions on H1(H3) are ignored. The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.
5. H2(H4) may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol, but never remains asserted longer than 4 clock cycles. Typically, a four clock cycle pulse is generated. But in the case where a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously. Thus, anytime after the leading edge of the H2(H4) pulse, new data can be entered in the PI/T double buffered input latches. The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.

A sample timing diagram is shown in Figure 4. The H2(H4) interlocked and pulsed input handshake protocols are shown. The DMAREQ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be 0 (refer to Port General Control Register); thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double buffered input transfers.

Double Buffered Output Transfers—The PI/T supports double buffered output transfers in all modes. Data, written by the bus master to the PI/T, is stored in the port's output latch. The peripheral accepts the data by asserting H1(H3), which causes the next data to be moved to the port's output latch as soon as it is available. The function of H2(H4) is programmable; it may indicate whether new

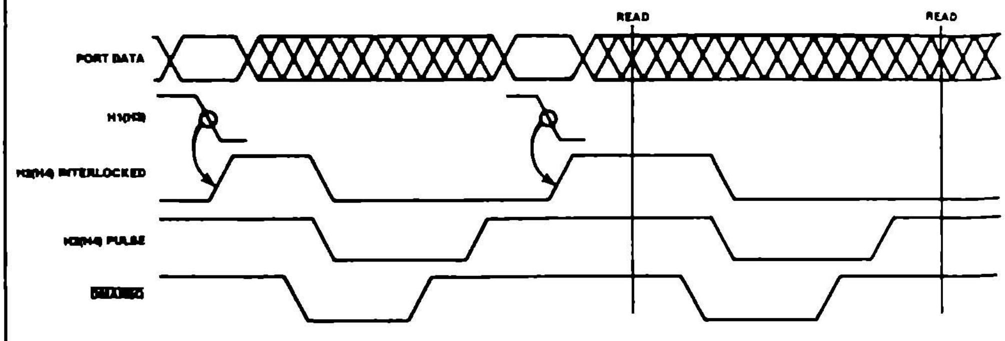


Figure 4. Double Buffered Input Transfers

data has been moved to the output latch or it may serve other purposes. The H1S(H3S) status bit may be programmed for two interpretations. Normally the status bit is a 1 when there is at least one latch in the double buffered data path that can accept new data. After writing one byte/word of data to the ports, an interrupt service routine could check this bit to determine if it could store another byte/word; thus, filling both latches. When the bus master is finished, it is often useful to be able to check whether all of the data has been transferred to the peripheral. The H1S(H3S) status control bit of the port A and B control registers provide this flexibility. The programmable options of the H2(H4) pin are given below, depending on the mode.

1. H2(H4) may be an edge sensitive input pin independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S (H4S) status bit is set. It is reset by the direct method (refer to Direct Method of Resetting Status), the RESET pin being asserted, or when the H12 enable (H34 enable) bit of the port general control register is 0.

2. H2(H4) may be a general purpose output pin that is always negated. The H2S(H4S) status bit is always 0.

3. H2(H4) may be a general purpose output pin that is always asserted. The H2S(H4S) status bit is always 0.

4. H2(H4) may be an output pin in the interlocked output handshake protocol. H2(H4) is asserted two clock cycles after data is transferred to the double buffered output latches. The data remains stable and H2(H4) remains asserted until the next asserted edge of the H1(H3) input. At that time, H2(H4) is asynchronously negated. As soon as the next data is available, it is transferred to the output latches. When H2(H4) is negated, asserted transitions on H1(H3) have no effect on the data paths. As is explained later, however, in modes 2 and 3 they do control the three-state output buffers of the bidirectional port(s). The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.

5. H2(H4) may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked output protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is generated. But in the case where a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously shortening the pulse. The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.

A sample timing diagram is shown in Figure 5. The H2(H4) interlocked and pulsed output handshake protocols are shown. The DMAREQ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be 0; thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double buffered output transfer.

Requesting Bus Master Service — The PI/T has several means of indicating a need for service by a bus master. First, the processor may poll the port status register. It contains a status bit for each handshake pin, plus a level bit that always reflects the instantaneous state of that handshake pin. A status bit is 1 when the PI/T needs servicing, i.e., generally when the bus master needs to read or write data to the ports, or when a handshake pin used as a simple status input has been asserted. The interpretation of these bits is dependent on the chosen mode and submode.

Second, the PI/T may be placed in the processor's interrupt structure. As mentioned previously, the PI/T contains port A and B control registers that configure the handshake pins. Other bits in these registers enable an interrupt associated with each handshake pin. This interrupt is made available through the PC5/PIRQ pin, if the PIRQ function is selected. Three additional conditions are required for PIRQ to be asserted: (1) the handshake pin status bit set, (2) the corresponding interrupt (service request) enable bit is set, (3) and DMA requests are not associated with that data transfer (H1 and H3 only). The

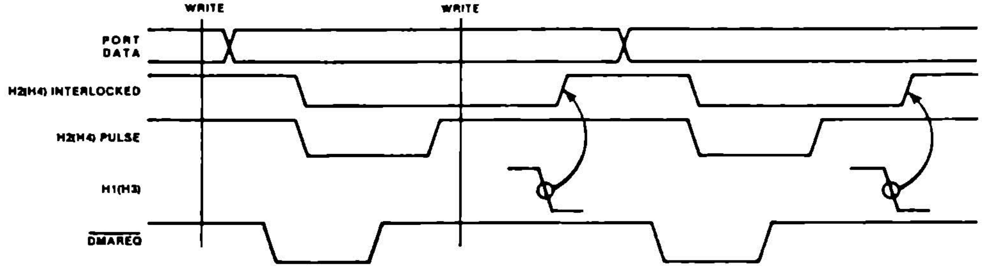


Figure 5. Double Buffered Output Transfers

conditions from each of the four handshake pins and corresponding status bits are ORed to determine PIRQ.

The third method of requesting service is via the PC4/DMAREQ pin. This pin can be associated with double buffered transfers in each mode. If it is used as a DMA controller request, it can initiate requests to keep the PI/T's input/output double buffering empty/full as much as possible. It will not overrun the DMA controller.

Vectored, Prioritized Port Interrupts—Use of SC68000 compatible vectored interrupts with the PI/T requires the PIRQ and PIACK pins. When PIACK is asserted, the PI/T places an 8-bit vector on the data pins D0-D7. Under normal conditions, this vector corresponds to highest priority, enabled, active port interrupt source with which the DMAREQ pin is not currently associated. The most significant six bits are provided by the port interrupt vector register (PIVR), with the lower two bits supplied by prioritization logic according to conditions present when PIACK is asserted. It is important to note that the only effect on the PI/T caused by interrupt acknowledge cycles is that the vector is placed on the data bus. Specifically, no registers, data, status, or other internal states of the PI/T are affected by the cycle.

Several conditions may be present when the PIACK input is asserted. These conditions affect the PI/T's response and the termination of the bus cycle. If the PI/T has no interrupt function selected, or is not asserting PIRQ, the PI/T will make no response to PIACK (DTACK will not be asserted). If the PI/T is asserting PIRQ when PIACK is received, the PI/T will output the contents of the PIVR and the prioritization bits. If the PIVR has not been initialized, $0F will be read from this register. These conditions are summarized in Table 4.

The vector table entries for the PI/T appear as a contiguous block of four vector numbers whose common upper six bits are programmed in the PIVR. The following list pairs each interrupt source with the 2-bit value provided by the prioritization logic, when interrupt acknowledge is asserted.

H1 source—00
H2 source—01
H3 source—10
H4 source—11

Autovectored Port Interrupts—Autovectored interrupts use only the PIRQ pin. The operation of the PI/T with vectored and autovectored interrupts is identical except that no vectors are supplied and the PC6/PIACK pin can be used as a port C pin.

Direct Method of Resetting Status—In certain modes one or more handshake pins can be used as edge sensitive inputs for the sole purpose of setting bits in the port status register. These bits consist of simple flip flops. They are set (to 1) by the occurrence of the asserted edge of the handshake pin input. Resetting a handshake status bit can be done by writing an 8-bit mask to the port status register. This is called the direct method of resetting. To reset a status bit that is resettable by the direct method, the mask must contain a 1 in the bit position of the port status register corresponding to the desired bit. Other positions must contain 0's. For status bits that are not resettable by the direct method in the chosen mode, the data written to the port status register has no effect. For status bits that are resettable by the direct method in the chosen mode, a 0 in the mask has no effect.

Handshake Pin Sense Control—The PI/T contains exclusive OR gates to control the

Table 4 RESPONSE TO PORT INTERRUPT ACKNOWLEDGE

Conditions	$\overline{\text{PIRQ}}$ negated OR interrupt request function not selected	$\overline{\text{PIRQ}}$ asserted
PIVR has not been initialized since $\overline{\text{RESET}}$	No response from PI/T No $\overline{\text{DTACK}}$	PI/T provides $0F, the Uninitialized Vector
PIVR has been initialized since $\overline{\text{RESET}}$	No response from PI/T No $\overline{\text{DTACK}}$	PI/T provides PIVR contents with prioritization bits

*The uninitialized vector is the value returned from an interrupt vector register before it has been initialized sense of each of the handshake pins, whether used as inputs or outputs. Four bits in the port general control register can be programmed to determine whether the pins are asserted in the low or high voltage state. As with other control registers, these bits are reset to 0 when the RESET pin is asserted, defaulting the asserted level to be low.

Enabling Ports A and B—Certain functions involved with double buffered data transfers, the handshake pins, and the status bits, can be disabled by the external system or by the programmer during initialization. The port general control register contains two bits, H12 enable and H34 enable, which control these functions. These bits are cleared to the 0 state when the RESET pin is asserted, and the functions are disabled. The functions are the following:

1. Independent of other actions by the bus master or peripheral (via the handshake pins), the PI/T's disabled handshake controller is held to the 'empty' state, i.e., no data is present in the double buffered data path.
2. When any handshake pin is used to set a simple status flip flop, unrelated to double buffered transfers, these flip flops are held reset to 0 (see Table 3).
3. When H2(H4) is used in an interlocked or pulsed handshake with H1(H3), H2(H4) is held negated, regardless of the chosen mode, submode, and primary direction. Thus, for double buffered input transfers, the programmer can signal a peripheral when the PI/T is ready to begin transfers by setting the associated handshake enable bit to 1.

The Port A and B Alternate Registers—In addition to the port A and B data registers, the PI/T contains port A and B alternate registers. These registers are read only, and simply provide the instantaneous level of each port pin. They have no effect on the operation of the handshake pins, double buffered transfers, status bits, or any other aspect of the PI/T, and they are mode/submode independent.

PORT MODES

Mode 0—Unidirectional 8-Bit Mode
In mode 0, ports A and B operate independently. Each can be configured in any of its three possible submodes:

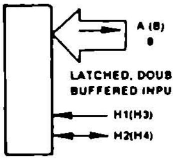
Submode 00—Double buffered input
Submode 01—Double buffered output
Submode 1X—Bit I/O

Handshake pins H1 and H2 are associated with port A and configured by programming the port A control register. (The H12 enable bit of the port general control register enables port A transfers.) Handshake pins H3 and H4 are associated with port B and configured by programming the port B control register. (The H34 enable bit of the port general control register enables port B transfers.) The port A and B data direction registers operate in all three submodes. Along with the submode, they affect the data read and written at the associated data register according to Table 5. They also enable the output buffer associated with each port pin. The DMAREQ pin may be associated with either (not both) port A or port B, but does not function if the bit I/O submode is programmed for the chosen port.

Port A or B Submode 00 (8-Bit Double Buffered Input)—In mode 0, double buffered input transfers of up to 8-bits are available by programming submode 00 in the desired port's control register. The operation of H2 and H4 can be selected by programming the port A and port B control registers, respectively. All five double buffered input handshake options, previously mentioned in the Port General Information and Conventions section, are available.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Output pins can be used independently of the input transfer. However, read bus cycles to the data register do remove data from the port. Therefore, care should be taken to avoid processor instructions that perform unwanted read cycles. Refer to Figure 4 for a sample timing diagram.

MODE 0 SUBMODE 00



Port A or B Submode 01 (8-bit Double Buffered Output)—In mode 0, double buffered output transfers of up to 8 bits are available by programming submode 01 in the desired port's control register. The operation of H2 and H4 can be selected by programming the port A and port B control registers, respectively. All five double buffered output handshake options, previously mentioned in the Port General Information and Conventions section, are available.

For pins used as inputs, data written to the associated data register is double buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled. Refer to Figure 5 for a sample timing diagram.

MODE 0 SUBMODE 01

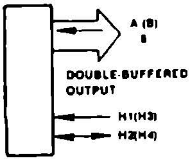


Table 5 MODE 0 PORT DATA PATHS

Mode	Read Port A/B Data Register		Write Port A/B Data Register	
	DDR = 0	DDR = 1	DDR = X	
0 Submode 00	FIL, D B	FOL Note 3	FOL, S B	Note 1
0 Submode 01	Pin	FOL Note 3	IOL/FOL, D B	Note 2
0 Submode 1X	Pin	FOL Note 3	FOL, S B	Note 1

Abbreviations
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S B – Single Buffered
D B – Double Buffered
DDR – Data Direction Register

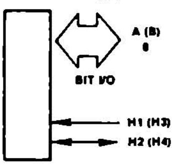
Note 1 Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 2 Data is latched in the double-buffered output data registers. The data in the final output latch will appear on the port pin if the DDR is a 1.
Note 3 The output drivers that connect the final output latch to the pins are turned on.

Preliminary

Port A or B Submode 1X (Bit I/O)—In mode 0, simple bit I/O is available by programming submode 1X in the desired port's control register. This submode is intended for applications in which several independent devices must be controlled or monitored. Data written to the associated data register is single buffered. If the data direction register bit for that pin is a 1 (output), the output buffer is enabled. If it is 0 (input), data written is still latched, but is not available at the pin. Data read from the data register is the instantaneous value of the pin or what was written to the data register, depending on the contents of the data direction register. H1(H3) is an edge sensitive status input pin only and it controls no data related function. The H1S(H3S) status bit is set following the asserted edge of the input waveform. It is reset by the direct method, the RESET pin being asserted, or when the H12 enable (H34 enable) bit is 0.

H2(H4) can be programmed as a simple status input (identical to H1(H3)), or as an asserted or negated output. The interlocked or pulsed handshake configurations are not available.

MODE 0 SUBMODE 1X



Mode 1—Unidirectional 16-Bit Mode
In mode 1, ports A and B are concatenated to form a single 16-bit port. The port B submode field controls the configuration of both parts. The possible submodes are:

- Port B submode X0—double buffered input
- Port B submode X1—double buffered output

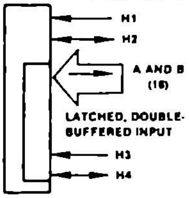
Handshake pins H3 and H4, configured by programming the port B control register, are associated with the 16-bit double buffered transfer. These 16-bit transfers are enabled by the H34 enable bit of the port general control register. Handshake pins H1 and H2 can be used as simple status inputs not related to the 16-bit data transfer or H2 can be an output. Enabling of the H1 and H2 handshake pins is done by the H12 enable bit of the port general control register. The port A and B data direction registers operate in each submode. Along with the submode, they affect the data read and written at the data register according to Table 6. They also enable the output buffer associated with each port pin. The DMAREQ pin can be associated only with H3.

Mode 1 can provide convenient, high speed 16-bit transfers. The port A and B data registers are addressed for compatibility with the SC68000 move peripheral (MOVEP) instruction and with the SC68450 DMAC. To take advantage of this, port A should contain the most significant byte of data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols are keyed to accesses to the port B data register in mode 1. If it is accessed last, the 16-bit double buffered transfers proceed smoothly.

Port B Submode X0 (16-Bit Double Buffered Input)—In mode 1 port B submode X0, double buffered input transfers of up to 16-bits can be obtained. The level of all 16 pins is asynchronously latched with the asserted edge of H3. The processor can check the H3S status bit to determine if new data is present. The DMAREQ pin can be used to signal a DMA controller to empty the input buffers. Regardless of the bus master, port A data should be read first. (Actually, port A data need not be read at all.) Port B data should be read last. The operation of the internal handshake controller, the H3S bit, and DMAREQ are keyed to the reading of the port B data register. (The SC68450 DMAC can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 can be programmed for all five of the handshake options mentioned in the Port General Information and Conventions section.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Thus, output pins can be used independently of the input transfer. However, read bus cycles to the port B data register do remove data, so care should be taken to avoid unwanted read cycles.

MODE 1 PORT B SUBMODE X0



Port B Submode X1 (16-Bit Double Buffered Output)—Refer to Figure 4 for a sample timing diagram. In mode 1 port B submode X1, double buffered output transfers of up to 16 bits can be obtained. Data is written by the bus master (processor or DMA controller) in two bytes. The first byte (most significant) is written to the port A data register. It is stored in a temporary latch until the next byte is written to the port B data register. Then all 16 bits are transferred to the final output latches

Table 6 MODE 1 PORT DATA PATHS

Mode	Read Port A/B Register		Write Port A/B Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
1, Port B Submode X0	FIL, D B	FOL Note 3	FOL, S B Note 2	FOL, S B Note 2
1, Port B Submode X1	Pin	FOL Note 3	IOL/FOL, D B, Note 1	IOL/FOL, D B, Note 1

Note 1 Data written to Port A goes to a temporary latch. When the Port B data register is later written, Port A data is transferred to IOL/FOL

Note 2 Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0

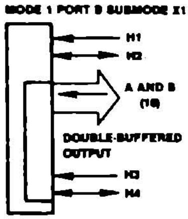
Note 3 The output drivers that connect the final output latch to the pins are turned on

Abbreviations

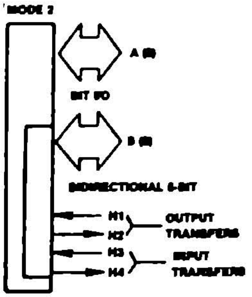
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S B – Single Buffered
D B – Double Buffered
DDR – Data Direction Register

of ports A and B. Both options for interpretation of the H3S status bit, mentioned in Port General Information and Comments section, are available and apply to the 16-bit port as a whole. The DMAREQ pin can be used to signal a DMA controller to transfer another word to the port output latches. (The SC68450 DMAC can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 can be programmed for all five of the handshake options mentioned in the Port General Information and Comments section.

For pins used as inputs, data written to either data register is double buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled. Refer to Figure 5 for a sample timing diagram.



Mode 2—Bidirectional 8-Bit Mode
In mode 2, port A is used for simple bit I/O with no associated handshake pins. Port B is used for bidirectional 8-bit double buffered transfers. H1 and H2, enabled by the H12 enable bit in the port general control register, control output transfers, while H3 and H4, enabled by the port general control register H34 enable bit, control input transfers. The instantaneous direction of the data is determined by the H1 handshake pin. The port B data direction register is not used. The port A and port B submode fields do not affect the PI/T operation in mode 2.



Double Buffered I/O (Port B)—The only aspect of bidirectional double buffered transfers that differs from the unidirectional modes lies in controlling the port B output buffers. They are controlled by the level of H1. When H1 is negated, the port B output buffers (all eight) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated in response to an asserted H2 which indicates that new output data is present in the double buffered latches. Following acceptance of the data, the peripheral asserts H1 disabling the port B output buffers. Other than controlling the output buffer, H1 is edge sensitive as in other modes. Input transfers proceed identically to the double buffered input protocol described in the Port General Information and Conventions Section. In mode 2, only the interlocked and pulsed handshake pin options are available on H2 and H4. The DMAREQ pin can be associated with either input transfers (H3) or output transfers (H1), but not both. Refer to Table 7 for a summary of the port B data register responses in mode 2.

Bit I/O (Port A)—Mode 2, port A performs simple bit I/O with no associated handshake pins. This configuration is intended for applications in which several independent devices must be controlled or monitored. Data written to the port A data register is single buffered. If the port A data direction register bit for that pin is 1 (output), the output buffer is enabled. If it is 0, data written is still latched but not available at the pin. Data read from the data register is either the instantaneous value of the pin or what was written to the data register, depending on the contents of the port A data direction register. This is summarized in table 8.

Mode 3—Bidirectional 16-Bit Double Buffered I/O
In mode 3, ports A and B are used for bidirectional 16-bit double buffered transfers. H1 and H2 control output transfers, while H3 and H4 control input transfers. (H1 and H2 are enabled by the H12 enable bit while H3 and H4 are enabled by the H34 enable bit of the port general control register.) The instantaneous direction of the data is determined by the H1 handshake pin, and thus, the data direction registers are not used. The port A and port B submode fields do not affect PI/T operation in mode 3.

The only aspect of bidirectional double buffered transfers that differs from the unidirectional modes lies in controlling the port A and B output buffers. They are controlled by the level of H1. When H1 is negated, the output buffers (all 16) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated in response to an asserted H2, which indicates that new output data is present in the double buffered latches. Following acceptance of the data, the peripheral asserts H1, disabling the output buffers. Other than controlling the output buffers H1 is edge sensitive as in other modes. Input transfers proceed identically to the double buffered input protocol described in the Port General Information and Conventions section. Port A and B data is latched with the asserted edge of H3 in

Table 7 MODE 2 PORT B DATA PATHS

Mode	Read Port B Data Register	Write Port B Data Register
2	FIL, D B	IOL/FOL, D B

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
D B – Double Buffered

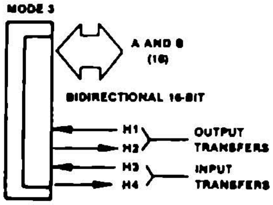
Table 8 MODE 2 PORT A DATA PATHS

Mode	Read Port A Data Register		Write Port A Data Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
2	Pin	FOL	FOL	FOL, S B

Abbreviations.
S.B – Single Buffered
FOL – Final Output Latch
DDR – Data Direction Register

mode 3, only the interlocked and pulsed handshake pin options are available to H2 and H4. The DMAREQ pin can be associated with either input transfers (H3) or output transfers (H1), but not both. H2 indicates when new data is available in the port B (and implicitly port A) output latches, but unless the buffer is enabled by H1, the data is not driving the pins.

Mode 3 can provide convenient high speed 16-bit transfers. The port A and B data registers are addressed for compatibility with the SC68000's move peripheral (MOVEP) instruction and with the SC68450 DMAC. To take advantage of this, port A should contain the most significant data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols are keyed to accesses to the port B data register in mode 3. If it is accessed last, the 16-bit double buffered transfer proceeds smoothly. Refer to Table 9 for a summary of the port A and B data paths in mode 3.


MODE 3
A AND B
(16)
BIDIRECTIONAL 16-BIT
H1
H2
OUTPUT TRANSFERS
H3
H4
INPUT TRANSFERS

DMA REQUEST OPERATION

The direct memory access request (DMAREQ) pulse (when enabled) is associated with output or input transfers to keep the initial and final output latches full or empty, respectively. Figures 6 and 7 show all the possible paths in generating DMA requests.

DMAREQ is generated on the bus side of the SC68230 by the synchronized[1] chip select. If the conditions of Figures 6 and 7 are met, an access of the bus (assertion of CS) will cause DMAREQ to be asserted three PI/T clocks (plus the delay time from the clock edge) after CS is synchronized. DMAREQ remains asserted for three clock cycles (plus the delay time from the clock edge) and is then negated.

Table 9 MODE 3 PORT A AND B DATA PATHS

<table>
<tr><th>Mode</th><th>Read Port A and B Data Register</th><th>Write Port A and B Data Register</th></tr>
<tr><td>3</td><td>FIL, D B</td><td>IOL/FOL, D B , Note 1</td></tr>
<tr><td colspan="3">Note 1 Data written to Port A goes to a temporary latch When the Port B data register is later written, Port A data is transferred to IOL/FOL</td></tr>
<tr><td colspan="3">Abbreviations
IOL – Initial Output Latch S B – Single Buffered
FOL – Final Output Latch D B – Double Buffered
FIL – Final Input Latch</td></tr>
</table>

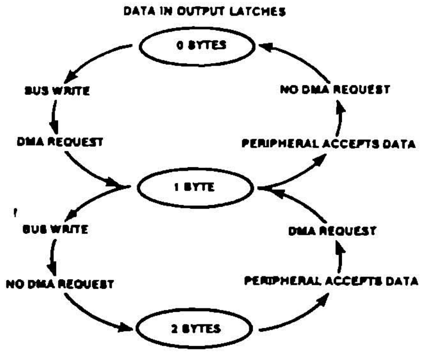

DATA IN OUTPUT LATCHES
0 BYTES
BUS WRITE
NO DMA REQUEST
DMA REQUEST
PERIPHERAL ACCEPTS DATA
1 BYTE
BUS WRITE
DMA REQUEST
NO DMA REQUEST
PERIPHERAL ACCEPTS DATA
2 BYTES

Figure 6. DMAREQ Associated with Output Transfers

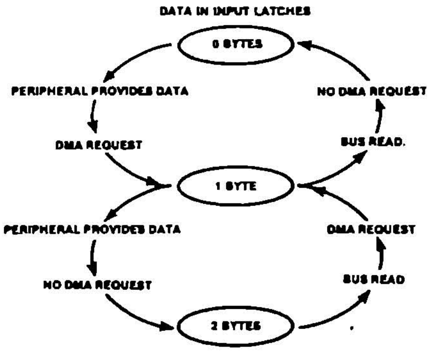

DATA IN INPUT LATCHES
0 BYTES
PERIPHERAL PROVIDES DATA
NO DMA REQUEST
DMA REQUEST
BUS READ.
1 BYTE
PERIPHERAL PROVIDES DATA
DMA REQUEST
NO DMA REQUEST
BUS READ
2 BYTES

Figure 7. DMAREQ Associated with Input Transfers

---

[1] Synchronized means that the input signal has been seen by the PI/T on the appropriate edge of the clock (rising edge for H1(H3) and falling edge for CS). (Refer to the Bus Interface section for the exception concerning CS.) If a bus access (assertion of CS) and a port access (assertion of H1(H3)) occur at the same time, CS will be recognized without delay. H1(H3) will be recognized one clock cycle later.

The DMAREQ pulse associated with a peripheral or port side of the PI/T is caused by the synchronized H1(H3) input. If the conditions of figures 6 and 7 are met, a port access (assertion of the H1(H3) input) will cause DMAREQ to be asserted 2.5 PI/T clock cycles (plus the delay time from clock edge) after H1(H3) is synchronized. DMAREQ remains asserted for three clock cycles (plus the delay time from the clock edge) and is then negated.

TIMER

The SC68230 timer can provide several facilities needed by S68000 operating systems. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. Also, it can be used for elapsed time measurement or as a device watchdog.

The PI/T timer contains a 24-bit synchronous down counter that is loaded from three 8-bit counter preload registers. The 24-bit counter can be clocked by the output of a 5-bit (divide by 32) prescaler or by an external timer input TIN. If the prescaler is used, it can be clocked by the system clock (CLK pin) or by the TIN external input. The counter signals the occurrence of an event primarily through zero detection. (A zero is when the value of the 24-bit timer is equal to zero.) This sets the zero detect status (ZDS) bit in the timer status register. It can be checked by the processor or can be used to generate a timer interrupt. The ZDS bit is reset by writing a 1 to the timer status register in that bit position.

The general operation of the timer is flexible and easily programmable. The timer is fully configured and controlled by programming the 8-bit timer control register. It controls the choice between the port C operation and the timer operation of three timer pins, whether the counter is loaded from the counter preload register or rolls over when zero detect is reached, the clock input, whether the prescaler is used, and whether the timer is enabled.

Run/Halt Definition

The overall operation of the timer is described in terms of the run or halt states. The control of the current state is determined by programming the timer control register. When in the halt state, all of the following occur:

1. The prior contents of the counter is not altered and is reliably readable via the count registers.
2. The prescaler is forced to $1F whether or not it is used.
3. The ZDS status bit is forced to 0, regardless of the possible zero contents of the 24-bit counter.

The run state is characterized by:

1. The counter is clocked by the source programmed in the timer control register.
2. The counter is not reliably readable.
3. The prescaler is allowed to decrement if programmed for use.
4. The ZDS status bit is set when the 24-bit counter transitions from $000001 to $000000.

Timer Rules

This following set of rules allow easy application of the timer.

1. Refer to the run/halt definition above.
2. When the RESET pin is asserted, all bits of the timer control register go to 0, configuring the dual function pins as port C inputs.
3. The contents of the counter preload registers and counter are not affected by the RESET pin.
4. The count registers provide a direct read data path from each portion of the 24-bit counter, but data written to their addresses is ignored. (This results in a normal bus cycle.) These registers are readable at any time, but their contents are never latched. Unreliable data may be read when the timer is in the run state.
5. The counter preload registers are readable and writable at any time and this occurs independently of any timer operation. No protection mechanisms are provided against ill-timed writes.
6. The input frequency to the 24-bit counter from the TIN pin or prescaler output must be between 0 and the input frequency at the CLK pin divided by 32 regardless of the configuration chosen.
7. For configurations in which the prescaler is used (with the CLK pin or TIN pin as an input), the contents of the counter preload register (CPR) is transferred to the counter the first time that the prescaler passes from $00 to $1F (rolls over) after entering the run state. Thereafter, the counter decrements or is loaded from the counter preload register when the prescaler rolls over.
8. For configurations in which the prescaler is not used, the contents of the counter preload registers are transferred to the counter on the first asserted edge of the TIN input after entering the run state. On subsequent asserted edges the counter decrements or is loaded from the counter preload registers.
9. The lowest value allowed in the counter preload register for use with the counter is $000001.

Timer Interrupt Acknowledge Cycles

Several conditions can be present when the timer interrupt acknowledge pin (TIACK) is asserted. These conditions affect the PI/T's response and the termination of the bus cycle (see Table 10).

PROGRAMMER'S MODEL

The internal accessible register organization is represented in Table 11. Address space within the address map is reserved for future expansion. Throughout the PI/T data sheet, the following conventions are maintained.

1. A read from a reserved location in the map results in a read from the 'null register'. The null register returns all zeros for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
2. Unused bits of a defined register are denoted by '*' and are read as zeros.
3. Bits that are unused in the chosen mode/submode but are used in others are denoted by 'X', and are readable and writable. Their content, however, is ignored in the chosen mode/submode

Table 10 RESPONSE TO TIMER INTERRUPT ACKNOWLEDGE

PC3/TOUT Function	Response to Asserted $\overline{\text{TIACK}}$
PC3 - Port C Pin	No response No $\overline{\text{DTACK}}$
TOUT Square Wave	No response No $\overline{\text{DTACK}}$
TOUT - Negated Timer Interrupt Request	No response. No $\overline{\text{DTACK}}$
TOUT - Asserted Timer Interrupt Request	Timer Interrupt Vector Contents $\overline{\text{DTACK}}$ Asserted

Preliminary

Table 11 PI/T REGISTER ADDRESSING ASSIGNMENTS

Register	Register Select Bits					Accessible	Affected by Reset	Affected by Read Cycle
	5	4	3	2	1			
Port General Control Register (PGCR)	0	0	0	0	0	R W	Yes	No
Port Service Request Register (PSRR)	0	0	0	0	1	R W	Yes	No
Port A Data Direction Register (PADDR)	0	0	0	1	0	R W	Yes	No
Port B Data Direction Register (PBDDR)	0	0	0	1	1	R W	Yes	No
Port C Data Direction Register (PCDDR)	0	0	1	0	0	R W	Yes	No
Port Interrupt Vector Register (PIVR)	0	0	1	0	1	R W	Yes	No
Port A Control Register (PACR)	0	0	1	1	0	R W	Yes	No
Port B Control Register (PBCR)	0	0	1	1	1	R W	Yes	No
Port A Data Register (PADR)	0	1	0	0	0	R W	No	**
Port B Data Register (PBDR)	0	1	0	0	1	R W	No	**
Port A Alternate Register (PAAR)	0	1	0	1	0	R	No	No
Port B Alternate Register (PBAR)	0	1	0	1	1	R	No	No
Port C Data Register (PCDR)	0	1	1	0	0	R W	No	No
Port Status Register (PSR)	0	1	1	0	1	R W*	Yes	No
Timer Control Register (TCR)	1	0	0	0	0	R W	Yes	No
Timer Interrupt Vector Register (TIVR)	1	0	0	0	1	R W	Yes	No
Counter Preload Register High (CPRH)	1	0	0	1	1	R W	No	No
Counter Preload Register Middle (CPRM)	1	0	1	0	0	R W	No	No
Counter Preload Register Low (CPRL)	1	0	1	0	1	R W	No	No
Count Register High (CNTRH)	1	0	1	1	1	R	No	No
Count Register Middle (CNTRM)	1	1	0	0	0	R	No	No
Count Register Low (CNTRL)	1	1	0	0	1	R	No	No
Timer Status Register (TSR)	1	1	0	1	0	R W*	Yes	No

*A write to this register may perform a special status resetting operation.
**Mode dependent.

R = Read
W = Write

4. All registers are addressable as 8-bit quantities. To facilitate operation with the MOVEP instruction and the DMAC, addresses are ordered such that certain sets of registers can also be accessed as words (2 bytes) or long words (4 bytes).

Port General Control Register (PGCR)
The port general register controls many of the functions that are common to the overall operation of the ports. The PGCR is composed of three major fields: bits 7 and 6 define the operational mode of ports A and B and affect operation of the handshake pins and status bits; bits 5 and 4 allow a software controlled disabling of particular hardware associated with the handshake pins of each port; and bits 3-0 define the sense of the handshake pins. The PGCR is always readable and writable. All bits are reset to 0 when the RESET pin is asserted.

The port mode control field should be altered only when the H12 enable and H34 enable bits are 0. Except when mode 0 is desired, the port general control register must be written once to establish the mode, and again to enable the respective operation(s).

Port General Control Register (PGCR)

7	6	5	4	3	2	1	0
Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense

7 6	Port Mode Control
0 0	Mode 0 (unidirectional 8-Bit mode)
0 1	Mode 1 (unidirectional 16-bit mode)
1 0	Mode 2 (bidirectional 8-bit mode)
1 1	Mode 3 (bidirectional 16-bit mode)

5 H34 Enable
0 Disabled
1 Enabled

4 H12 Enable
0 Disabled
1 Enabled

3-0 Handshake Pin Sense
0 The associated pin is at the high voltage level when negated and at the low voltage level when asserted.
1 The associated pin is at the low voltage level when negated and at the high voltage level when asserted.

Port Service Request Register (PSRR)
The port service request register controls other functions that are common to the overall operation to the ports. It is composed of four major fields: bit 7 is unused and is always read as 0; bits 6 and 5 define whether interrupt or DMA requests are generated from activity on the H1 and H3

PARALLEL INTERFACE/TIMER

SC68230

Preliminary

handshake pins; bits 4 and 3 determine whether two dual function pins operate as port C or port interrupt request/acknowledge pins; and bits 2, 1, and 0 control the priority among all port interrupt sources. Since bits 2, 1, and 0 affect interrupt operation, it is recommended that they be changed only when the affected interrupt(s) is disabled or known to remain inactive. The PSRR is always readable and writable.

All bits are reset to 0 when the RESET pin is asserted.

Port Service Request Register (PSRR)

7	6	5	4	3	2	1	0
*	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control		

6 5 SVCRQ Select

0 X The PC4/DMAREQ pin carries the PC4 function; DMA is not used.

1 0 The PC4/DMAREQ pin carries the DMAREQ function and is associated with double buffered transfers controlled by H1. H1 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain DMAREQ pulses, port A control register bit 1 (H1 SVCRQ enable) must be a 1.

1 1 The PC4/DMAREQ pin carries the DMAREQ function and is associated with double buffered transfers controlled by H3. H3 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain DMAREQ pulses, Port B Control Register bit 1 (H3 SVCRQ Enable) must be 1.

4 3 Interrupt Pin Function Select

0 0 The PC5/PIRQ pin carries the PC5 function.
The PC6/PIACK pin carries the PC6 function.

0 1 The PC5/PIRQ pin carries the PIRQ function.
The PC6/PIACK pin carries the PC6 function.

1 0 The PC5/PIRQ pin carries the PC5 function.
The PC6/PIACK pin carries the PIACK function.

1 1 The PC5/PIRQ pin carries the PIRQ function.
The PC6/PIACK pin carries the PIACK function.

Bits 2, 1, and 0 determine port interrupt priority. The priority is shown in descending order left to right.

Port Interrupt Priority Control

2 1 0	Highest			Lowest
0 0 0	H1S	H2S	H3S	H4S
0 0 1	H2S	H1S	H3S	H4S
0 1 0	H1S	H2S	H4S	H3S
0 1 1	H2S	H1S	H4S	H3S
1 0 0	H3S	H4S	H1S	H2S
1 0 1	H3S	H4S	H2S	H1S
1 1 0	H4S	H3S	H1S	H2S
1 1 1	H4S	H3S	H2S	H1S

Port A Data Direction Register (PADDR)
The port A data direction register determines the direction and buffering characteristics of each of the port A pins. One bit in the PADDR is assigned to each pin. A 0 indicates that the pin is used as an input, while a 1 indicates it is used as an output. The PADDR is always readable and writable. This register is ignored in mode 3.

All bits are reset to the 0 (input) state when the RESET pin is asserted.

Port B Data Direction Register (PBDDR)
The PBDDR is identical to the PADDR for the port B pins and the port B data register, except that this register is ignored in modes 2 and 3.

Port C Data Direction Register (PCDDR)
The port C data direction register specifies whether each dual function pin that is chosen for the port C operation is an input (0) or an output (1) pin. The PCDDR, along with bits that determine the respective pin's function, also specify the exact hardware to be accessed at the port C data register address. (See the Port C Data Register description for more details.) The PCDDR is an 8-bit register that is readable and writable at all times. Its operation is independent of the chosen PI/T mode.

These bits are cleared to 0 when the RESET pin is asserted.

Port Interrupt Vector Register (PIVR)
The port interrupt vector register contains the upper order six bits of the four port interrupt vectors. The contents of this register can be read two ways: by an ordinary read cycle, or by a port interrupt acknowledge bus cycle. The exact data read depends on how the cycle was initiated and other factors. Behavior during a port interrupt acknowledge cycle is summarized in Table 4.

Port Interrupt Vector Register (PIVR)

7	6	5	4	3	2	1	0
Interrupt Vector Number						*	*

From a normal read cycle (CS), there is never a consequence to reading this register. Following negation of the RESET pin, but prior to writing to the PIVR, a $0F will be read. After writing to the register, the upper 6 bits may be read and the lower 2 bits are forced to 0. No prioritization computation is performed.

Port A Control Register (PACR)
The port A control register, in conjunction with the programmed mode and the port B submode, control the operation of port A and the handshake pins H1 and H2. The port A control register contains five fields: bits 7 and 6 specify the port A submode, bits 5, 4, and 3 control the operation of the H2 handshake pin and H2S status bit; bit 2 determines whether an interrupt will be generated when the H2S status bit goes to 1; bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H1S status bit. The PACR is always readable and writable.

All bits are cleared to 0 when the RESET pin is asserted. When the port A submode field is relevant in a mode/submode definition, it must not be altered unless the H12 enable bit in the port general control register is 0 (see Table 3).

The operation of H1 and H2 and their related status bits is given below for each of the modes specified by the port general control register bits 7 and 6.

Bits 2 and 1 carry the same meaning in each mode/submode, and thus are specified only once.

Port A Control Register (PACR)

7	6	5	4	3	2	1	0
Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl

2 H2 Interrupt Enable

0 The H2 interrupt is disabled.

1 The H2 interrupt is enabled.

1 H1 SVCRQ Enable

0 The H1 interrupt and DMA request are disabled.

1 The H1 interrupt and DMA request are enabled.

Mode 0, Port A Submode 00

5 4 3 H2 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked input handshake protocol.
1 1 1 Output pin—pulsed input handshake protocol.

0 H1 Status Control
X Not used

Mode 0, Port A Submode 01

5 4 3 H2 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked output handshake protocol.
1 1 1 Output pin—pulsed output handshake protocol.

0 H1 Status Control
0 The H1S status bit is 1 when either the port A initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H1S status bit is 1 when both of the port A output latches are empty. It is 0 when at least one latch is full.

Mode 0, Port A Submode 1X

5 4 3 H2 Control
0 X X Input pin—status only.
1 X 0 Output pin—always negated.
1 X 1 Output pin—always asserted.

0 H1 Status Control
X Not used.

Mode 1, Port A Submode XX, Port B Submode X0

5 4 3 H2 Control
0 X X Input pin—status only.
1 X 0 Output pin—always asserted.
1 X 1 Output pin—always asserted.

0 H1 Status Control
X Not used.

PACR Mode 1 Port A Submode XX Port B Submode X1

5 4 3 H2 Control
0 X X Input pin—status only.
1 X 0 Output pin—always negated.
1 X 1 Output pin—always asserted.

0 H1 Status Control
X Not used.

Mode 2

5 4 3 H2 Control
X X 0 Output pin—interlocked output handshake protocol.
X X 1 Output pin—pulsed output handshake protocol.

0 H1 Status Control
0 The H1S status bit is 1 when either the port B initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H1S status bit is 1 when both of the port B output latches are empty. It is 0 when at least one latch is full.

Mode 3

5 4 3 H2 Control
X X 0 Output pin—interlocked output handshake protocol.
X X 1 Output pin—pulsed output handshake protocol.

0 H1 Status Control
0 The H1S status bit is 1 when either the initial or final output latch of port A and B can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H1S status bit is 1 when both the initial and final output latches of ports A and B are empty. It is 0 when either the initial or final latch of ports A and B is full.

Port B Control Register (PBCR)
The port B control register specifies the operation of port B and the handshake pins H3 and H4. The port B control register contains five fields: bits 7 and 6 specify the port B submode; bits 5, 4, and 3 control the operation of the H4 handshake pin and H4S status bit; bit 2 determines whether an interrupt will be generated when the H4S status bit goes to 1; bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H3S status bit. The PACR is always readable and writable. There is never a consequence to reading the register.

All bits are cleared to 0 when the RESET pin is asserted. When the port B submode field is relevant in a mode/submode definition, it must not be altered unless the H34 enable bit in the port general control register is 0 (see Table 3).

The operation of H3 and H4 and their related status bits is given below for each of the modes specified by the port general control register bits 7 and 6.

Bits 2 and 1 carry the same meaning in each mode/submode, and thus are specified only once.

Port B Control Register (PBCR)

7	6	5	4	3	2	1	0
Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl

2 H4 Interrupt Enable
0 The H4 interrupt is disabled.
1 The H4 interrupt is enabled.

1 H3 SVCRQ Enable
0 The H3 interrupt and DMA request are disabled.
1 The H3 interrupt and DMA request are enabled.

Mode 0, Port B Submode 00

5 4 3 H4 Control
0 X X Input pin — status only.
1 0 0 Output pin — always negated.
1 0 1 Output pin — always asserted.
1 1 0 Output pin — interlocked input handshake protocol.
1 1 1 Output pin — pulsed input handshake protocol.

0 H3 Status Control
0 Not used.

Mode 0, Port B Submode 01

5 4 3 H4 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked output handshake protocol.
1 1 1 Output pin—pulsed output handshake protocol.

0 H3 Status Control
0 The H3S status bit is 1 when either the port B initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H3S status bit is 1 when both of the port B output latches are empty. It is 0 when at least one latch is full.

Preliminary

Mode 0, Port B Submode 1X

5 4 3 H4 Control
0 X X Input pin—status only
1 X 0 Output pin—always negated
1 X 1 Output pin—always asserted

0 H3 Status Control
X Not used

Mode 1, Port B Submode X0

5 4 3 H4 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked input handshake protocol
1 1 1 Output pin—pulsed input handshake protocol.

0 H3 Status Control
X Not used.

Mode 1, Port B Submode X1

5 4 3 H4 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked output handshake protocol.
1 1 1 Output pin—pulsed output handshake protocol.

0 H3 Status Control
0 The H3S status bit is 1 when either the initial or final output latch of port A and B can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H3S status bit is 1 when both the initial and final output latches of ports A and B are empty. It is 0 when neither the initial or final latch of ports A and B is full.

Mode 2

5 4 3 H4 Control
X X 0 Output pin—interlocked input handshake protocol
X X 1 Output pin—pulsed input handshake protocol.

0 H3 Status Control
X Not used.

Mode 3

5 4 3 H4 Control
X X 0 Output pin—interlocked input handshake protocol.
X X 1 Output pin—pulsed input handshake protocol.

0 H3 Status Control
X Not used

Port A Data Register (PADR)
The port A data register is an address for moving data to and from the port A pins. The port A data direction register determines whether each pin is an input (0) or an output (1), and is used in configuring the actual data paths. PADR is mode dependent.

This register is readable and writable at all times. Depending on the chosen mode/submode, reading or writing may affect the double buffered handshake mechanism. The port A data register is not affected by the assertion of the RESET pin.

Port B Data Register (PBDR)
The port B data register is an address for moving data to and from the port B pins. The port B data direction register determines whether each pin is an input (0) or an output (1), and is used in configuring the actual data paths. PBDR is mode dependent.

This register is readable and writable at all times. Depending on the chosen mode/submode, reading or writing may affect the double buffered handshake mechanism. The port B data register is not affected by the assertion of the RESET pin.

Port A Alternate Register (PAAR)
The port A alternate register is an alternate address for reading the port A pins. It is a read only address and no other PI/T condition is affected. In all modes, the instantaneous pin level is read and no input latching is performed except at the data bus interface (see Bus Interface Connection). Writes to this address are answered with DTACK, but the data is ignored.

Port B Alternate Register (PBAR)
The port B alternate register is an alternate address for reading the port B pins. It is a read only address and no other PI/T condition is affected. In all modes, the instantaneous pin level is read and no input latching is performed except at the data bus interface (see Bus Interface Connection). Writes to this address are answered with DTACK, but the data is ignored.

Port C Data Register (PCDR)
The port C data register is an address for moving data to and from each of the eight port C/alternate function pins. The exact hardware accessed is determined by the type of bus cycle (read or write) and individual conditions affecting each pin. These conditions are: whether the pin is used for the port C or alternate function and whether the port C data direction register indicates the input or output direction. The port C data register is single buffered for output pins and not buffered for input pins. These conditions are summarized in table 12.

The Port C data register is not affected by the assertion of the RESET pin.

The operation of the PCDR is independent of the chosen PI/T mode.

Note that two additional useful benefits result from this structure. First, it is possible to directly read the state of a dual function pin while used for the non port C function. Second, it is possible to generate program controlled transitions on alternate function pins by switching back to the port C function, and writing to the PCDR.

This register is readable and writable at all times.

Port Status Register (PSR)
The port status register contains information about handshake pin activity. Bits 7-4 show the instantaneous level of the respective handshake pin, and is independent of the handshake pin sense bits in the port general control register. Bit 3-0

Table 12 PCDR HARDWARE ACCESSES

Read Port C Data Register			
Port C function PCDDR = 0	Port C function PCDDR = 1	Alternate function PCDDR = 0	Alternate function PCDDR = 1
pin	Port C output register	pin	Port C output register
Write Port C Data Register			
Port C function PCDDR = 0	Port C function PCDDR = 1	Alternate function PCDDR = 0	Alternate function PCDDR = 1
Port C output register, buffer disabled	Port C output register, buffer enabled	Port C output register	Port C output register

are the respective status bits referred to throughout this data sheet. Their interpretation depends on the programmed mode/submode of the PI/T. For bits 3-0, a 1 is the active or asserted state.

Port Status Register (PSR)

7	6	5	4	3	2	1	0
H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S

Timer Control Register (TCR)
The timer control register determines all operations of the timer. Bits 7-5 configure the PC3/TOUT and PC7/TIACK pins for port C, square wave, vectored interrupt, or autovectored interrupt operation; bit 4 specifies whether the counter receives data from the counter preload register or continues counting when zero detect is reached; bit 3 is unused and is read as 0; bits 2 and 1 configure the path from the CLK and TIN pins to the counter controller; bit 0 enables the timer. This register is readable and writable at all times.

All bits are cleared to 0 when the RESET pin is asserted.

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TOUT/TIACK Control			ZD Ctrl	•	Clock Control		Timer Enable

7 6 5 TOUT/TIACK Control

0 0 X The dual function pins PC3/TOUT and PC7/TIACK carry the port C function.

0 1 X The dual function pin PC3/TOUT carries the TOUT function. In the run state it is used as a square wave output and is toggled on zero detect. The TOUT pin is high while in the halt state. The dual function pin PC7/TIACK carries the PC7 function.

1 0 0 The dual function pin PC3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual function pin PC7/TIACK carries the TIACK function; however, since interrupt request is negated, the PI/T produces no response, i.e., no data or DTACK to an asserted TIACK. Refer to the Timer Interrupt Cycle section for details. This combination and the 101 state below support vectored timer interrupts.

1 0 1 The dual function pin PC3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is 1. The dual function pin PC7/TIACK carries the TIACK function and is used as a timer interrupt acknowledge input. Refer to the Timer Interrupt Acknowledge Cycle section for details. This combination and the 100 state above support vectored timer interrupts.

1 1 0 The dual function pin PC3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual function pin PC7/TIACK carries the PC7 function.

1 1 1 The dual function pin PC3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is 1. The dual function pin PC7/TIACK carries the PC7 function and autovectored interrupts are supported.

4 Zero Detect Control

0 The counter is loaded from the counter preload register on the first clock to the 24-bit counter after zero detect, and resumes counting.

1 The counter rolls over on zero detect, then continues counting.

Bit 3 is unused and is always read as 0.

2 1 Clock Control

0 0 The PC2/TIN input pin carries the port C function and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer enable bit determines whether the timer is in the run or halt state.

0 1 The PC2/TIN pin serves as a timer input and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer is in the run state when the timer enable bit is 1 and the TIN pin is high; otherwise the timer is in the halt state.

1 0 The PC2/TIN pin serves as a timer input and the prescaler is used. The prescaler is decremented following the rising transition of the TIN pin after syncing with the internal clock. The 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer enable bit determines whether the timer is in the run or halt state.

1 1 The PC2/TIN pin serves as a timer input and the prescaler is unused. The 24-bit counter is decremented or loaded from the counter preload registers following the rising edge of the TIN pin after syncing with the internal clock. The timer enable bit determines whether the timer is in the run or halt state.

0 Timer Enable

0 Disabled.

1 Enabled.

Timer Interrupt Vector Register (TIVR)
The timer interrupt vector register contains the 8-bit vector supplied when the timer interrupt acknowledge pin TIACK is asserted. The register is readable and writable at all times, and the same value is always obtained from a normal read cycle and a timer interrupt acknowledge bus cycle (TIACK). When the RESET pin is asserted, the value of $0F is automatically loaded into the register. Refer to the Timer Interrupt Acknowledge Cycle section for more details.

Counter Preload Register H,M,L (CPRH-L)
The counter preload registers are a group of three 8-bit registers used for storing data to be transferred to the counter. Each of the registers is individually addressable, or the group may be accessed with the MOVEP.L or the MOVEP.W instructions. The address one less than the address of CPRH is the null register, and is reserved so that zeros are read in the upper 8 bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

The registers are readable and writable at all times. A read cycle proceeds independently of any transfer to the counter, which may be occurring simultaneously.

To insure proper operation of the PI/T timer, a value of $000000 cannot be stored in the counter preload registers for use with the counter.

The RESET pin does not affect the contents of these registers.

Counter Preload Register H, M, L (CPRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CPRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CPRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CPRL

Count Register H, M, L (CNTRH-L)
The count registers are a group of three 8-bit addresses at which the counter can be read. The contents of the counter are not latched during a read bus cycle; thus, the data read at these addresses is not guaranteed if the timer is in the run state. (Bits 2, 1 and 0 of the timer control register specify the state.) Write operations to these addresses result in a normal bus cycle but the data is ignored.

Each of the registers is individually addressable, or the group can be accessed with the MOVEP.L or the MOVEP.W instructions. The address one less than the address of CNTRH is the null register, and is reserved so that zeros are read in the upper 8 bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

Count Register H, M, L (CNTRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CNTRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CNTRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CNTRL

Timer Status Register (TSR)
The timer status register contains one bit from which the zero detect status can be determined. The ZDS status bit (bit 0) is an edge sensitive flip flop that is set to 1 when the 24-bit counter decrements from $000001 to $000000. The ZDS status bit is cleared to 0 following the direct clear operation (similar to that of the ports), or when the timer is halted. Note also that when the RESET pin is asserted, the timer is disabled and thus enters the halt state.

This register is always readable without consequence. A write access performs a direct clear operation if bit 0 in the written data is 1. Following that, the ZDS bit is 0.

This register is constructed with a reset dominant S-R flip flop so that all clearing conditions prevail over the possible zero detect condition.

Bits 7-1 are unused and are read as 0.

Timer Status Register (TSR)

7	6	5	4	3	2	1	0
•	•	•	•	•	•	•	ZDS

TIMER APPLICATIONS SUMMARY

Periodic Interrupt Generator
In this configuration the timer generates a periodic interrupt. The TOUT pin is connected to the system's interrupt request circuitry and the TIACK pin can be used as an interrupt acknowledge input to the timer. The TIN pin can be used as a clock input.

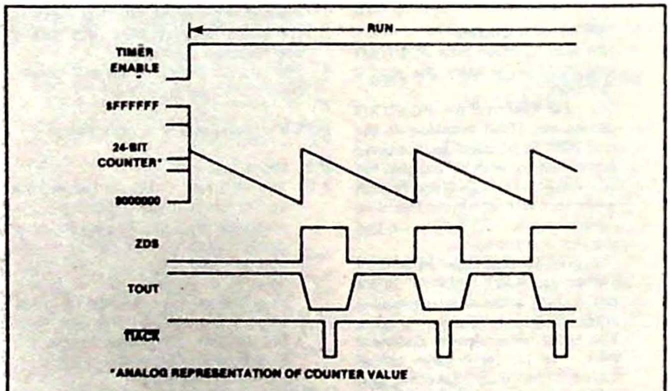
The processor loads the counter preload registers and timer control register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000, the ZDS status bit is set and the TOUT (interrupt request) pin is asserted. At the next clock to the 24-bit counter, it is again loaded with the contents of the CPRs and thereafter decrements. In normal operation, the processor must direct clear the status bit to negate the interrupt request (see Figure 8).

Periodic Interrupt Generator

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z.D. Ctrl	•	Clock Control		Timer Enable
1	X	1	0	X	00 or 1X		1

Square Wave Generator
In this configuration the timer produces a square wave at the TOUT pin. The TOUT pin is connected to the user's circuitry and the TIACK pin is not used. The TIN pin may be used as a clock input.

The processor loads the counter preload registers and timer control register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000, the ZDS status bit is set and the TOUT (square wave output) pin is toggled. At the next clock to the 24-bit counter, it is again loaded with the contents of the CPRs and thereafter decrements. In this application there is no need for the processor to direct clear the ZDS status bit; however, it is possible for the processor to sync itself with the square wave by clearing the ZDS status bit, then polling it. The processor can also read the TOUT level at the port C address.



*ANALOG REPRESENTATION OF COUNTER VALUE

Figure 8. Periodic Interrupt Generator

Preliminary

Note that the PC3/TOUT pin functions as PC3 following the negation of RESET. If used in the square wave configuration, a pullup resistor may be required to keep a known level prior to programming. Prior to enabling the timer, TOUT is high (see Figure 9).

Square Wave Generator

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z.D. Ctrl	•	Clock Control		Timer Enable
0	1	X	0	X	00 or 1X		Changed

Interrupt After Timeout

In this configuration the timer generates an interrupt after a programmed time period has expired. The TOUT pin is connected to the system's interrupt request circuitry and the TIACK pin can be an interrupt acknowledge input to the timer. The TIN pin may be used as a clock input.

This configuration is similar to the periodic interrupt generator except that the zero detect control bit is set. This forces the counter to roll over after zero detect is reached, rather than reloading from the CPRs. When the processor takes the interrupt, it can halt the timer and read the counter. This allows the processor to measure the delay time from zero detect (interrupt request) to entering the service routine. Accurate knowledge of the interrupt latency may be useful in some applications (see Figure 10).

Interrupt After Timeout

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z.D. Ctrl	•	Clock Control		Timer Enable
1	X	1	1	0	00 or 1X		Changed

Elapsed Time Measurement

Elapsed time measurement takes several forms; two are described below. The first configuration allows time interval measurement by software. No timer pins are used.

The processor loads the counter preload registers (generally with all 1s) and timer control register, and then enables the timer. The counter decrements until the ending event takes place. When it is desired to read the time interval, the processor must halt the timer, then read the counter. For applications in which the interval could have exceeded that programmable in this timer, interrupts can be counted to provide the equivalent of additional timer bits. At the end, the timer can be halted and read (see Figure 11).

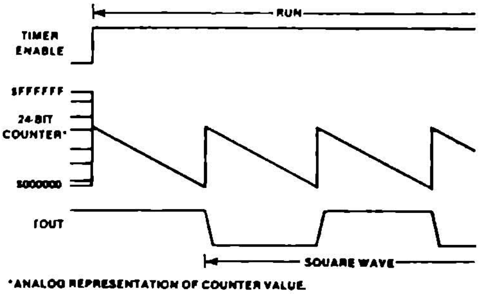


Figure 9. Square Wave Generator

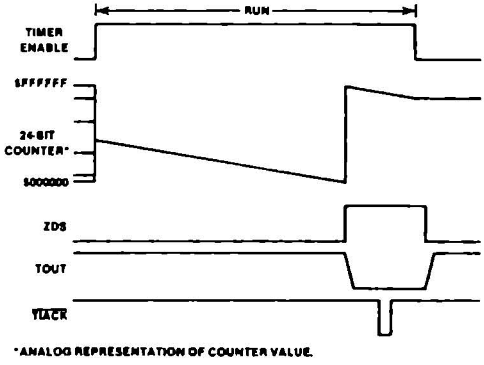
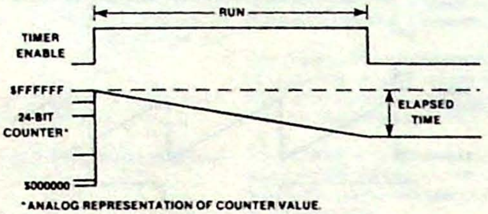
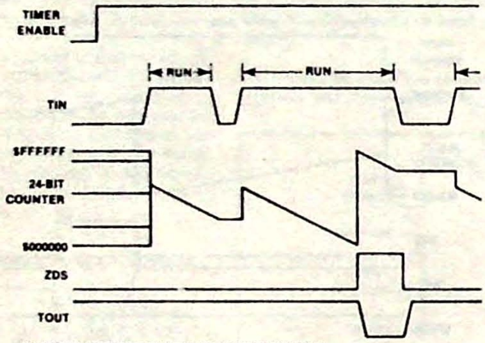


Figure 10. Single Interrupt After Timeout



*ANALOG REPRESENTATION OF COUNTER VALUE.

Figure 11. Elapsed Time Measurement



*ANALOG REPRESENTATION OF COUNTER VALUE.

Figure 12. Device Watchdog

System Clock

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z D Ctrl	*	Clock Control		Timer Enable
0	0	X	1	0	0	0	[illegible]

The second configuration allows measurement (counting) of the number of input pulses occurring in an interval in which the counter is enabled. The TIN input pin provides the input pulses. Generally the TOUT and TIACK pins are not used.

This configuration is identical to the elapsed time measurement/system clock configuration except that the TIN pin is used to provide the input frequency. It can be connected to a simple oscillator, and the same methods could be used. Alternately, it could be gated off and on externally and the number of cycles occurring while in the run state can be counted. However, minimum pulse width high and low specifications must be met.

External Clock

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z D Ctrl	*	Clock Control		Timer Enable
0	0	X	1	0	[illegible]	[illegible]	[illegible]

Device Watchdog

This configuration provides the watchdog function needed in many systems. The TIN pin is the timer input whose period at the high (1) level is to be checked. Once allowed by the processor, the TIN input pin controls the run/halt mode. The TOUT pin is connected to external circuitry requiring notification when the TIN pin has been asserted longer than the programmed time. The TIACK pin (interrupt acknowledge) is only needed if the TOUT pin is connected to interrupt circuitry.

The processor loads the counter preload register and timer control register, and then enables the timer. When the TIN input is asserted (1, high) the timer transfers the contents of the counter preload register to the counter and begins counting. If the TIN input is negated before zero detect is reached, the TOUT output and the ZDS status bit remain negated. If zero detect is reached while the TIN input is still asserted, the ZDS status bit is set and the TOUT output is asserted. (The counter rolls over and keeps on counting).

In either case, when the TIN input is negated, the ZDS status bit is 0, the TOUT output is negated, the counting stops, and prescaler is forced to all 1s (see Figure 12).

Device Watchdog

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z D Ctrl	*	Clock Control		Timer Enable
1	x	1	1	0	0	1	[illegible]

BUS INTERFACE CONNECTION

The PI/T has an asynchronous bus interface, primarily designed for use with the SC68000 microprocessor. With care, however, it can be connected to synchronous microprocessor buses. This section completely describes the PI/T's bus interface, and is intended for the asynchronous bus designer unless otherwise specified.

In an asynchronous system, the PI/T CLK can operate at a significantly different frequency, either higher or lower than the bus master and other system components, as long as all bus specifications are met. The SC68230 CLK pin has the same specifications as the SC68000 CLK, and must not be gated off at any time.

The following signals generate normal read and write cycles to the PI/T: CS (chip select), R/W (read/write), RS1-RS5 (five register select bits), D0-D7 (the 8-bit bidirectional data bus), and DTACK (data transfer acknowledge). To generate interrupt acknowledge cycles, PC6/PIACK or PC7/TIACK is used instead of CS and the register select pins are ignored. No combination of the following pins can be asserted simultaneously: CS, PIACK, or TIACK.

Read Cycles via Chip Select
This category includes all register reads, except port or timer interrupt acknowledge cycles. When CS is asserted, the register select and R/W inputs are latched internally. They must meet small setup and hold time requirements with respect to the asserted edge of CS (see the AC Electrical Characteristics table). The PI/T is not protected against aborted (shortened) bus cycles generated by an address error or bus error exception in which it is addressed.

Certain operations triggered by normal read (or write) bus cycles are not complete within the time allotted to the bus cycle. One example is transfers to/from the double buffered latches that occur as a result of the bus cycle. If the bus master's CLK is significantly faster than the PI/T's, the possibility exists that, following the bus cycle, CS can be negated then reasserted before completion of these internal operations. In this situation the PI/T does not recognize the reassertion of CS until these operations are complete. Only at that time does it begin the internal sequencing necessary to react to the asserted CS. Since CS also controls the DTACK response, this 'bus cycle recovery time' can be related to the CLK edge on which DTACK is asserted for that cycle. The PI/T will recognize the subsequent assertion of CS three CLK periods after the CLK edge on which DTACK was previously asserted.

The register select and R/W inputs pass through an internal latch that is transparent when the PI/T can recognize a new CS pulse (see above paragraph). Since the internal data bus of the PI/T is continuously enabled for read transfers, the read access time (to the data bus buffers) begins when the register selects are stabilized internally. Also, when the PI/T is ready to begin a new bus cycle, the assertion of CS enables the data bus buffers within a short propagation delay. This does not contribute to the overall read access time, unless CS is asserted significantly after the register select and R/W inputs are stabilized (as may occur with synchronous bus microprocessors).

In addition to chip select's previously mentioned duties, it controls the assertion of DTACK and latching of read data at the data bus interface. Except for controlling input latches and enabling the data bus buffers, all of these functions occur only after CS has been recognized internally and synchronized with the internal clock. Chip select is recognized on the falling edge of the CLK if the setup time is met, and DTACK is asserted (low) on the next falling edge of the CLK. Read data is latched at the PI/T's data bus interface at the same time DTACK is asserted. It is stable as long as chip select remains asserted independent of other external conditions.

From the above discussion, it is clear that if the CS setup time prior to the falling edge of the CLK is met the PI/T can consistently respond to a new read or write bus cycle every four CLK cycles. This fact is especially useful in designing the PI/T's clock in synchronous bus systems not using DTACK. (An extra CLK period is required in interrupt acknowledge cycles, see Read Cycles via Interrupt Acknowledge).

In asynchronous bus systems in which the PI/T's CLK differs from that of the bus master, generally there is no way to guarantee that the CS setup time with respect to the PI/T CLK is met. Thus, the only way to determine that the PI/T recognized the assertion of CS is to wait for the assertion of DTACK. In this situation, all latched bus inputs to the PI/T must be held stable until DTACK is asserted. These include register select, R/W, and write data inputs (see below).

System specifications impose a maximum delay from the trailing (negated) edge of chip select to the negated edge of DTACK. As system speeds increase, this becomes more difficult to meet with a simple pullup resistor tied to the DTACK line. Therefore, the PI/T provides an internal active pullup device to reduce the rise time, and a level sensitive circuit that later turns this device off. DTACK is negated asynchronously as fast as possible following the rising edge of chip select, then three-stated to avoid interference with the next bus cycle.

The system designer must take care that DTACK is negated and three-stated quickly enough after each bus cycle to avoid interference with the next one. With the SC68000 this necessitates a relatively fast external path from the data strobe to CS going negated.

Write Cycles
In many ways write cycles are similar to normal read cycles (see above). On write cycles, data at the D0-D7 pins must meet the same setup specifications as the register select and R/W lines. Like these signals, write data is latched on the asserted edge of CS, and must meet small setup and hold time requirements with respect to that edge. The same bus cycle recovery conditions exist as for normal read cycles. No other differences exist.

Read Cycles via Interrupt Acknowledge
Special internal operations take place on PI/T interrupt acknowledge cycles. The port interrupt vector register or the timer interrupt vector register are implicitly addressed by the assertion of PC6/PIACK or PC7/TIACK, respectively. The signals are first synchronized with the falling edge of the CLK. One clock period after they are recognized, the data bus buffers are enabled and the vector is driven onto the bus. DTACK is asserted after another clock period to allow the vector some setup prior to DTACK. DTACK is negated, then three-stated as with the normal read or write cycle when PIACK or TIACK is negated.

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

Parameter	Rating	Unit
Supply voltage	−0.3 to +7.0	V
Input voltage[3]	−0.3 to +7.0	V
Operating temperature range[2]	0 to +70	°C
Storage temperature	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0V ±5%, T_A = 0°C to +70°C[4,5])

	Parameter	Test Conditions	Limits		Unit
			Min	Max	
V_{IH}	Input high voltage		2.0	V_{CC}	V
V_{IL}	Input low voltage		−0.3	0.8	V
I_{in}	Input leakage current H1,H3,R/W,RESET,CLK, RS1-RS5, CS	V_{in} = 0 to 5.25V		10.0	µA
I_{TSI}	Three-state (off state) input current DTACK,PC0-PC7, D0-D7	V_{in} = 0.4 to 2.4		±20	µA
	H2,H4, PA0-PA7,PB0-PB7		−0.1	−1.0	mA
V_{OH}	Output high voltage DTACK, D0-D7	I_{LOAD} = −400mA, V_{CC} = min	2.4		V
	H2,H4, PB0-PB7, PA0-PA7	I_{LOAD} = −150mA, V_{CC} = min	2.4		V
	PC0-PC7	I_{LOAD} = −100mA, V_{CC} min	2.4		V
V_{OL}	Output low voltage PC3/TOUT,PC5/PIRQ	I_{LOAD} = 8.8mA, V_{CC} = min		0.5	V
	D0-D7, DTACK	I_{LOAD} = 5.3mA, V_{CC} = min		0.5	V
	PA0-PA7, PB0-PB7,H2,H4, PC0-PC2,PC4,PC6,PC7	I_{LOAD} = 2.4mA, V_{CC} = min		0.5	V
P_{INT}	Power dissipation	T_A = 0°C		500	mW
C_{in}	Input capacitance	V_{in} = 0V, T_A = 25°C, f = 1MHz		15	pF

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 30°C/W junction to ambient for ceramic package.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.

PARALLEL INTERFACE/TIMER SC68230

Preliminary

AC ELECTRICAL SPECIFICATIONS V_{CC} = 5VDC ± 5%, V_{SS} = 0VDC, T_A = 0°V to 70°C (see figures 14-18)[4,5]

Number	Characteristic	Tentative Limits				Unit
		6MHz		8MHz		
		Min	Max	Min	Max	
1	R/W, RS1-RS5 valid to CS low (setup time)	0		0		ns
2[15]	CS low to R/W and RS1-RS5 invalid (hold time)	100		65		ns
3[6]	CS low to CLK low (setup time)	30		20		ns
4[7]	CS low to data out valid (delay)		75		60	ns
5	RS1-RS5, R/W valid to data out valid (delay)		140		100	ns
6	CLK low to DTACK low (read/write cycle) (delay)	0	70	0	60	ns
7[8]	DTACK low to CS high (hold time)	0		0		ns
8	CS or PIACK or TIACK high to data out invalid (hold time)	0		0		ns
9	CS or PIACK or TIACK high to D0-D7 high impedance (delay)		50		45	ns
10	CS or PIACK or TIACK high to DTACK high (delay)		50		30	ns
11	CS or PIACK or TIACK high to DTACK high impedance (delay)		100		55	ns
12	Data In valid to CS low (setup time)	0		0		ns
13	CS low to data in invalid (hold time)	100		65		ns
14	Input data valid to H1(H3) asserted (setup time)	100		60		ns
15	H1(H3) asserted to input data invalid (hold time)	20		20		ns
16	Handshake input H1(H4) pulse width asserted	40		40		ns
17	Handshake input H1(H4) pulse width negated	40		40		ns
18	H1(H3) asserted to H2(H4) negated (delay)		150		120	ns
19	CLK low to H2(H4) asserted (delay)		100		100	ns
20[9]	H2(H4) asserted to H1(H3) asserted	0		0		ns
21[10]	CLK low to H2(H4) pulse negated (delay)		125		125	ns
22[14,16]	Synchronized H1(H3) to CLK low on which DMAREQ is asserted (see figures 6 and 7)	2.5	3.5	2.5	3.5	clk per
23	CLK low DMAREQ is asserted to CLK low on which DMAREQ is negated	3	3	3	3	clk per
24	CLK low to output data valid (delay) (modes 0, 1)		150		120	ns
25[14,16]	Synchronized H1(H3) to output data invalid (modes 0, 1)	1.5	2.5	1.5	2.5	clk per

[6] This specification only applies if the PI/T had completed all operations initiated by the previous bus cycle when CS was asserted. Following a normal read or write bus cycle all operations are complete within three CLKs after the falling edge of the CLK pin on which DTACK was asserted. If CS is asserted prior to completion of these operations the new bus cycle, and hence DTACK is postponed.

If all operations of the previous bus cycle were complete when CS was asserted, this specification is made only to insure that DTACK is asserted with respect to the falling edge of the CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the CS setup time is violated, DTACK may be asserted as shown, or may be asserted one clock cycle later.

[7] Assuming the RS1 RS5 to data valid time has also expired.

[8] This specification imposes a lower bound on CS low time, guaranteeing that CS will be low for at least 1 CLK period.

[9] This specification assures recognition of the asserted edge of H1(H3).

[10] This specification applies only when a pulsed handshake option is chosen and the pulse is not shortened due to an early asserted edge of H1(H3).

[11] CLK refers to the actual frequency of the CLK pin, not the maximum allowable CLK frequency.

[12] If the setup time on the rising edge of the clock is violated, H1(H3) may not be recognized until the next rising edge of the clock.

[13] This limit applies to the frequency of the signal at TIN compared to the frequency of the CLK signal during each clock cycle. If any period of the waveform at TIN is smaller than the period of the CLK signal at that instant, then it is likely that the timer circuit will completely ignore one cycle of the TIN signal.

If these two signals are derived from different sources, they will have different instantaneous frequency variations. In this case the frequency applied to the TIN pin must be distinctly less than the frequency at the CLK pin to avoid lost cycles of the TIN signal. With signals derived from different crystal oscillators applied to the TIN and CLK pins with fast rise and fall times, the TIN frequency can approach 80 to 90% of the frequency of the CLK signal without a loss of a cycle of the TIN signal.

If these two signals are derived from the same frequency source then the frequency of the signal applied to TIN can be 100% of the frequency at the CLK pin. They may be generated by different buffers from the same signal or one may be an inverted version of the other. The TIN signal may be generated by an AND function of the clock and a control signal.

[14] The maximum value is caused by a peripheral access (H1(H3) asserted) and bus access (CS asserted) occurring at the same time.

[15] See Bus Interface Connection for exception.

[16] Synchronized means that the input signal has been seen by the PI/T on the appropriate edge of the clock (rising edge for H1(H3) and falling edge for CS. Refer to Bus Interface Connection for exception concerning CS.

Preliminary

AC Electrical Specifications (Continued) V_{CC} = 5VDC ± 5%, V_{SS} = 0VDC, T_A = 0°V to 70°C (see figures 14-18)[4,5]

Number	Characteristic	Tentative Limits				Unit
		6MHz		8MHz		
		Min	Max	Min	Max	
26	H1 negated to output data valid (modes 2, 3)		70		50	ns
27	H1 asserted to output data high impedance (modes 2, 3)	0	70	0	70	ns
28	Read data valid to DTACK low (setup time)	0	0			ns
29	CLK low to data output valid (interrupt acknowledge cycle)		120		100	ns
30[12]	H1(H3) asserted to CLK high (setup time)	50		40		ns
31	PIACK or TIACK low to CLK low (setup time)	50		40		ns
32[16]	Synchronized CS to CLK low on which DMAREQ is asserted (see figures 6 and 7)	3	3	3	3	clk per
33[14,16]	Synchronized H1(H3) to CLK low on which H2(H4) is asserted	3.5	4.5	3.5	4.5	clk per
34	CLK low to DTACK low (interrupt acknowledge cycle) (delay)		75		75	ns
35	CLK low to DMAREQ low (delay)	0	120	0	100	ns
36	CLK low to DMAREQ high (delay)	0	120	0	100	ns
A	CLK low to PIRQ low or high impedance		250		225	ns
B[13]	TIN frequency (external clock)—prescaler used	0	1	0	1	Fclk(Hz)[11]
C	TIN frequency (external clock)—prescaler not used	0	1/32	0	1/32	Fclk(Hz)[11]
D	TIN pulse width high or low (external clock) ·	55		45		ns
E	TIN pulse width low (run/halt control)	1		1		clk
F	CLK low to TOUT high, low, or high impedance	0	250	0	225	ns
G	CS, PIACK, or TIACK high to CS, PIACK, or TIACK low	50		30		ns

CLOCK TIMING (see Figure 13)

Parameter		8MHz		8MHz		Unit
		Min	Max	Min	Max	
f	Frequency of operation	2.0	8.0	2.0	10.0	MHz
t_{cyc}	Cycle time	125	500	100	500	ns
t_{CL}	Clock pulse width	55	250	45	250	ns
t_{CH}	Clock pulse width	55	250	45	250	ns
t_{Cr}	Clock rise time		10		10	ns
t_{Cf}	Clock fall time		10		10	ns

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from

$T_J = T_A + (P_D \bullet \theta_{JA})$ (1)

Where

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is

$P_D = K - (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives

$K = P_D \bullet (T_A - 273°C) + \theta_{JA} \bullet P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

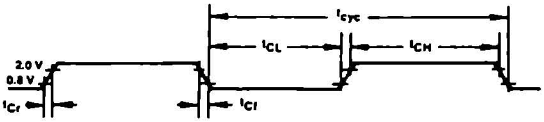


Figure 13. Input Clock Waveform

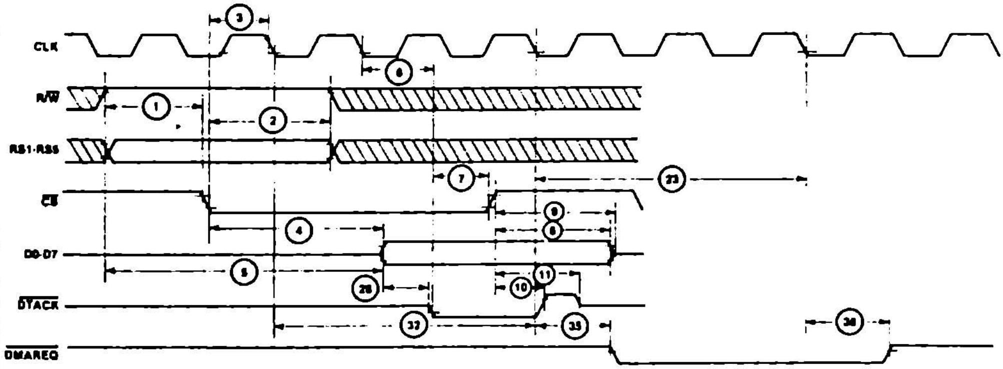


Figure 14. Bus Read Cycle Timing

Preliminary

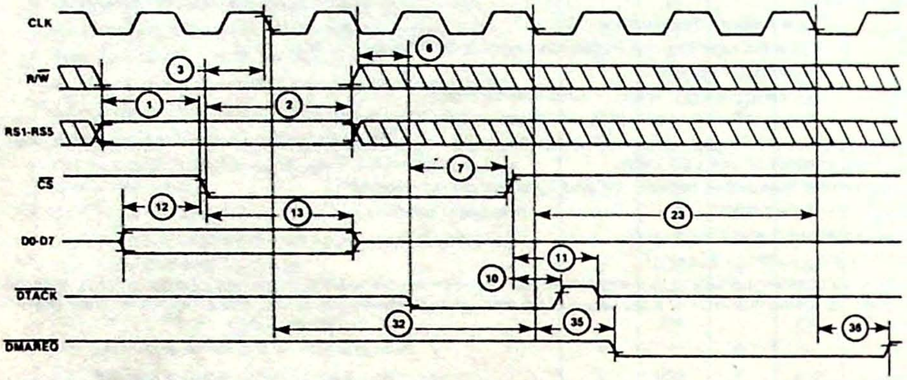

CLK
R/$\overline{W}$
RS1-RS5
$\overline{CS}$
D0-D7
$\overline{DTACK}$
$\overline{DMAREQ}$
1
2
3
6
7
10
11
12
13
23
32
35
36

Figure 15. Bus Write Cycle Timing

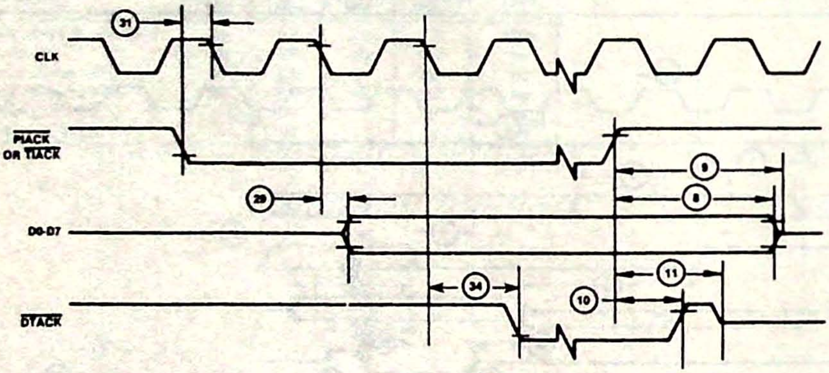

CLK
$\overline{PIACK}$ OR $\overline{TIACK}$
D0-D7
$\overline{DTACK}$
31
29
9
8
34
11
10

Figure 16. Interrupt Acknowledge Functional Timing

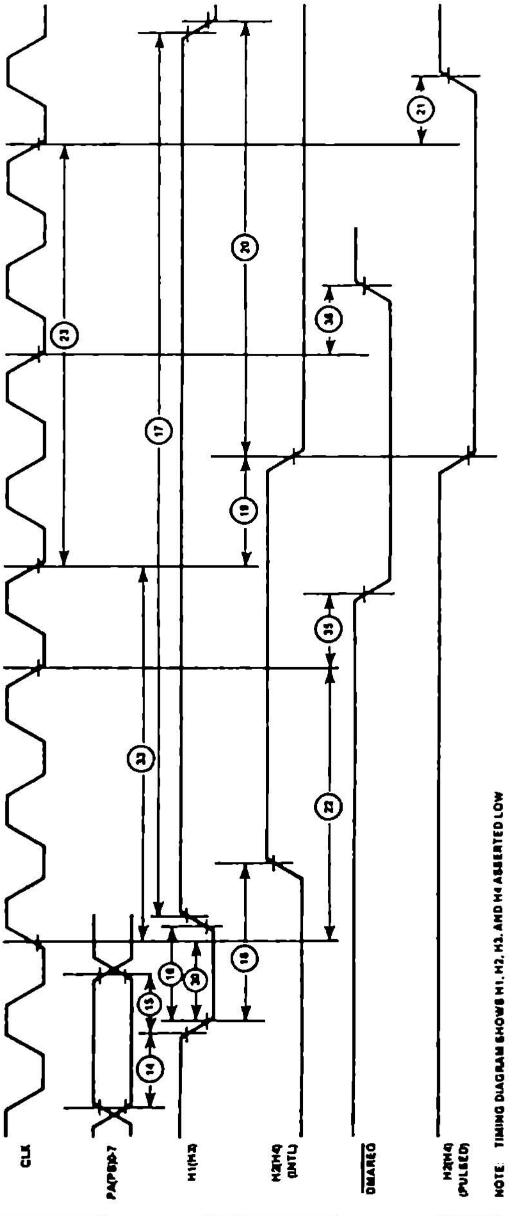
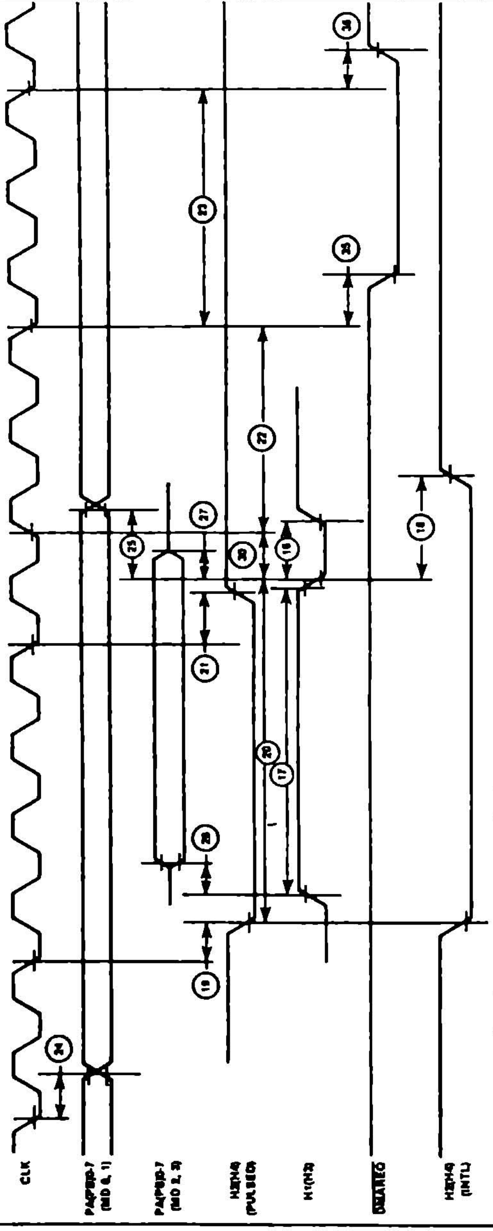

CLK
PA(PB)0-7
H1(H3)
H2(H4) (INTL)
DMAREQ
H2(H4) (PULSED)
14
15
16
30
18
33
17
19
20
23
22
35
36
21
NOTE: TIMING DIAGRAM SHOWS H1, H2, H3, AND H4 ASSERTED LOW

Figure 17. Peripheral Interface Input Timing

Preliminary


CLK
PA(PB)0-7 (MD 0, 1)
PA(PB)0-7 (MD 2, 3)
H2(H4) (PULSED)
H1(H3)
DMAREQ
H2(H4) (INTL)
34
19
28
20
17
21
30
25
27
16
22
18
23
35
36

NOTE: TIMING DIAGRAM SHOWS H1, H2, H3, AND H4 ASSERTED LOW.

Figure 18. Peripheral Interface Output Timing

DIRECT MEMORY ACCESS INTERFACE (DMAI) SC68430

PRODUCT BRIEF

DESCRIPTION

The SC68430 DMAI is a single channel DMA interface circuit which is intended to complement the performance and architectural capabilities of the SC68000 microprocessor. The DMAI functions by transferring a series of operands (data) between memory and a device: operand sizes may be byte, word, or long word. A block is a sequence of operands: the number of operands in the block is determined by a transfer count stored within the DMAI. The SC68430 can be programmed to utilize single cycle (cycle stealing) or burst data transfers.

The DMAI provides two interfaces. The microprocessor interface is fully compatible with the SC68000 microprocessor. The device interface includes lines for requesting, acknowledging, controlling, and timing the data transfers.

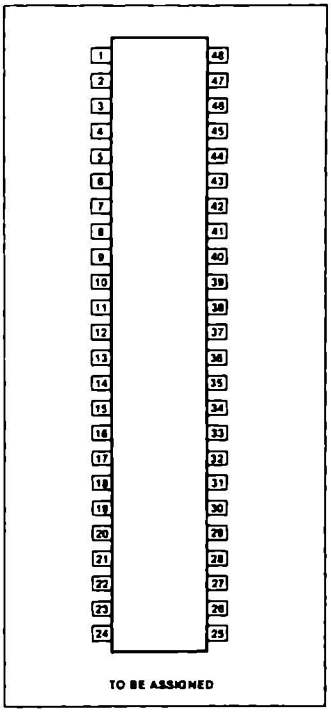
The DMAI may be considered a single-channel subset of the SC68450 four-channel DMA controller (DMAC). It is software compatible with the DMAC and provides similar interfacing signals to both the system bus and the device.

The SC68430 is constructed using Signetics ISL bipolar technology and is contained in a 48-pin dual-in-line package.

FEATURES

- **Single channel DMA controller**
- **Bus compatible with SC68000 microprocessor**
- **Software compatible with SC68450 DMA controller**
- **Single address transfers**
- **Cycle steal and burst mode operation**
- **Bus arbitration daisy chain**
- **Automatic rerun on bus error**
- **Supports SC68000 vectored interrupts**
- **Supports 32 bit transfers on VME bus**
- **24-bit address register**
- **16-bit transfer count register**
- **16 data/address lines (multiplexed) and 7 bidirectional address lines**
- **Maximum transfer rate of 4Mbytes per second**
- **Signetics ISL bipolar technology**
- **48-pin DIP**

PIN CONFIGURATION



OPERATION

The DMAI is used for efficient transfer of data between a peripheral device controller and the microprocessor memory. To transfer a block of data, the user initializes the memory address counter to specify the source or destination of the operands, and the memory transfer counter to specify the number of operands to be transferred. Each operand transfer requires only one bus cycle since the device is implicitly addressed. Transfers can be programmed to occur in a burst mode or in a cycle stealing mode. In the first case, the DMAI arbitrates for the bus and, when it obtains the bus, retains ownership until the transfer of the block of data is completed. In the second case, the DMAI transfers only a single operand and then relinquishes ownership of the bus unless a new transfer request is received from the device before the first transfer is completed.

Operand sizes can be specified as byte, word (16-bits), double word (2 x 16-bits), or long word (32-bits). The DMAI updates the address counter for each transfer and generates the appropriate data strobes depending on the operand size chosen.

The operation of the DMAI is determined by the programming of internal registers which are under software control of the system MPU. The structure of the registers is identical to that of the SC68450 4-channel DMA Controller, so that a program written to operate the DMAI will also operate the DMAC with no alteration. This allows expansion of system capabilities with minimal software impact. Registers contained within the DMAI are:

- — Channel status register
- — Channel error register
- — Device control register
- — Operation control register
- — Sequence control register
- — Channel control register
- — Interrupt vector register
- — Channel priority register
- — Memory transfer counter
- — Memory address counter

The DMAI can be programmed to interrupt the MPU upon the completion of the transfer of a block of data. Upon receipt of an interrupt acknowledge, the DMAI will output the contents of the interrupt vector register on the data bus for use by the MPU in a vectored interrupt system. A normal termination will output the contents of the IVR as loaded by the programmer. If the operation terminated due to an error, the least significant bit of the vector is modified by the DMAI to indicate an abnormal DMA termination.

MEMORY MANAGEMENT UNIT

SC68451

PRODUCT BRIEF

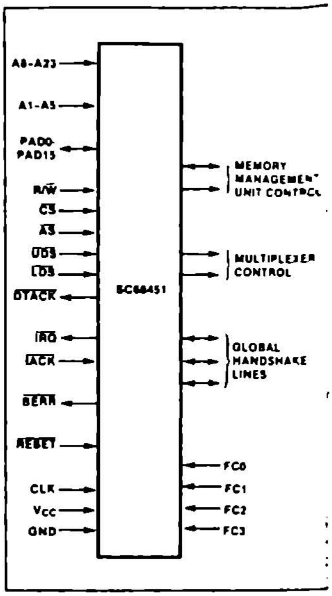
DESCRIPTION

The SC68451 Memory Management Unit (MMU) provides address translation and protection of the 16-megabyte addressing space of the SC68000. The MMU can be accessed by any potential bus master, such as instruction set processors, or DMA controllers. Each bus master (or processor) in the SC68000 family provides a function code and an address during each bus cycle. The function code specifies an address space while the address specifies a location within that address space. The function codes are provided by the SC68000 to distinguish between program and data spaces as well as supervisor and user spaces. This separation of address spaces provides the basis of protection in an operating system. By simplifying the programming model of the address space, the MMU also increases the reliability of a complex multiprocess system.

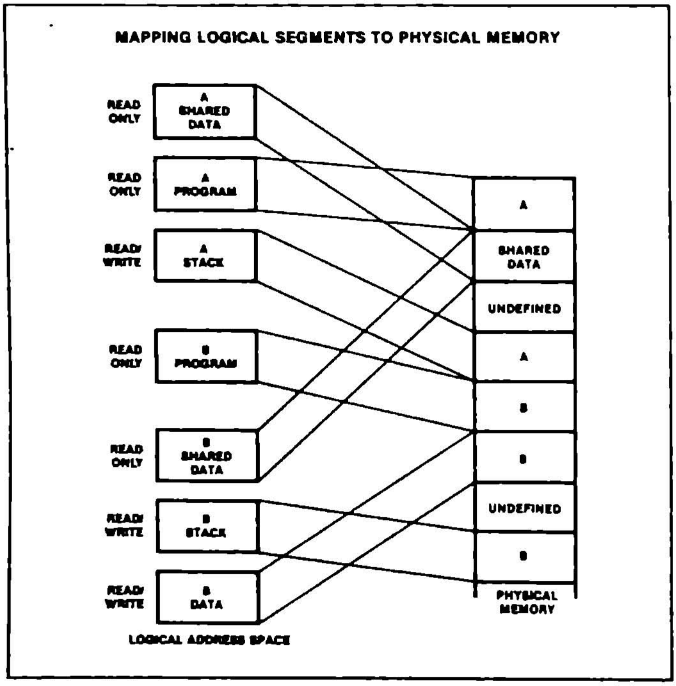
FEATURES

- **Separates address spaces of system and user resources**
- **Provides write protection**
- **Increases system reliability**
- **Provides efficient memory allocation**
- **Allows interprocess communication through shared resources**
- **Simplifies programming model of address space**
- **Minimizes operating system overhead with quick context switches**
- **32 segments with variable segment sizes**
- **Multiple MMU system capability**
- **Supports both paging and segmentation**
- **DMA compatible**
- **Provides virtual memory support**
- **SC68000 bus compatible**

FUNCTIONAL DIAGRAM


A8-A23
A1-A5
PAD0-PAD15
R/$\overline{W}$
$\overline{CS}$
$\overline{AS}$
$\overline{UDS}$
$\overline{LDS}$
$\overline{DTACK}$
$\overline{IRQ}$
$\overline{IACK}$
$\overline{BERR}$
$\overline{RESET}$
CLK
V_{CC}
GND
SC68451
MEMORY MANAGEMENT UNIT CONTROL
MULTIPLEXER CONTROL
GLOBAL HANDSHAKE LINES
FC0
FC1
FC2
FC3

MAPPING LOGICAL SEGMENTS TO PHYSICAL MEMORY


READ ONLY
A SHARED DATA
READ ONLY
A PROGRAM
READ/ WRITE
A STACK
READ ONLY
B PROGRAM
READ ONLY
B SHARED DATA
READ/ WRITE
B STACK
READ/ WRITE
B DATA
LOGICAL ADDRESS SPACE
A
SHARED DATA
UNDEFINED
A
B
B
UNDEFINED
B
PHYSICAL MEMORY

MARCH 1982

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC68681

Preview

DESCRIPTION

The Signetics SC68681 Dual Universal Asynchronous Receiver/Transmitters (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It is compatible with other S68000 family devices, and can also interface easily with other microprocessors. The DUART can be used in polled or interrupt driven systems.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

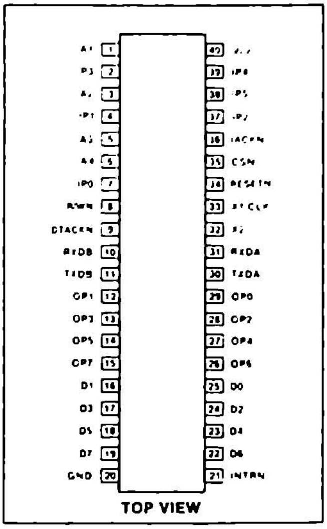
Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

Also provided on the SC68681 are a multipurpose 8-bit input port and an multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

FEATURES

- SC68000 bus compatible
- Dual full-duplex asynchronous receiver/transmitter
- Quadruple buffered receiver data registers
- Programmable data format
  - —5 to 8 data bits plus parity
  - —Odd, even, no parity or force parity
  - —1, 1.5 or 2 stop bits programmable in 1/16 bit increments
- Programmable baud rate for each receiver and transmitter selectable from:
  - —18 fixed rates: 50 to 38.4K baud
  - —One user defined rate derived from programmable timer/counter
  - —External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - —Normal (full duplex)
  - —Automatic echo
  - —Local loopback
  - —Remote loopback
- Multi-function programmable 16-bit counter/timer
- Multi-function 8-bit input port
  - —Can serve as clock or control inputs
  - —Change of state detection on four inputs
- Multi-function 8-bit output port
  - —Individual bit set/reset capability
  - —Outputs can be programmed to be status/interrupt signals
- Versatile interrupt system
  - —Single interrupt output with eight maskable interrupting conditions
  - —Interrupt vector output on interrupt acknowledge
  - —Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs
- Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec
- Automatic wake-up mode for multidrop applications
- Start-end break interrupt/status
- Detects break which originates in the middle of a character
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply

PIN CONFIGURATION



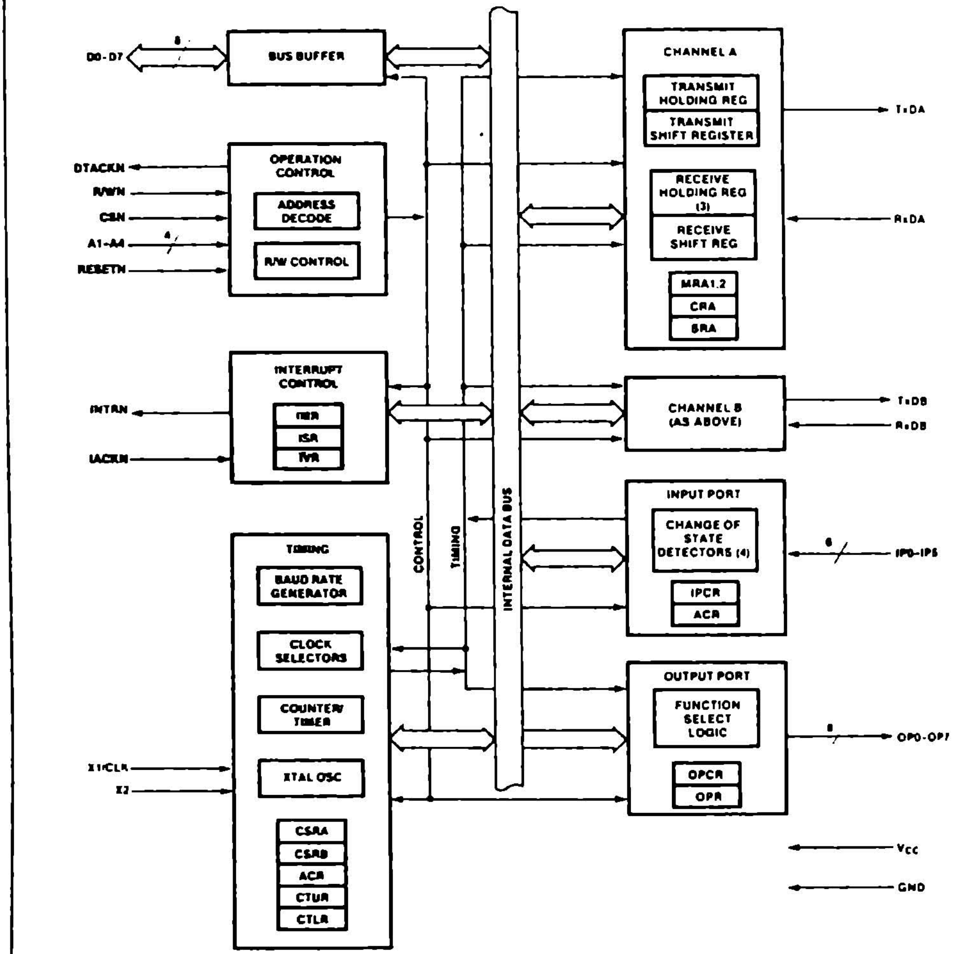
ORDERING CODE

PACKAGES	$V_{CC} = 5V \pm 5\%$ $T_A = 0°C$ to $70°C$
Ceramic DIP	SC68681CSI40
Plastic DIP	SC68681CSN40

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC68681

Preview

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
DTACKN
R/WN
CSN
A1-A4
4
RESETN
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTRN
IACKN
INTERRUPT CONTROL
IMR
ISR
IVR
X1/CLK
X2
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/TIMER
XTAL OSC
CSRA
CSRB
ACR
CTUR
CTLR
CONTROL
TIMING
INTERNAL DATA BUS
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
MRA1,2
CRA
SRA
TxDA
RxDA
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
CHANGE OF STATE DETECTORS (4)
IPCR
ACR
6
IP0-IP5
OUTPUT PORT
FUNCTION SELECT LOGIC
OPCR
OPR
8
OP0-OP7
Vcc
GND

MARCH 1982

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC68681

Preview

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0-D7	25-22, 16-19	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit
CSN	35	I	**Chip Select:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the R/WN and A1-A4 inputs. When high, places the D0-D7 lines in the 3-state condition.
R/WN	8	I	**Read/Write:** A high input indicates a read cycle and a low input indicates a write cycle, when a cycle is initiated by assertion of the CSN input.
A1-A4	1, 3, 5, 6	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESETN	34	I	**Reset:** A low clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), initializes the IVR to hex 0F, puts OP0-OP7 in the high state, stops the counter/timer, and puts channel A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
DTACKN	9	O	**Data Transfer Acknowledge:** Three-state active low output asserted in write, read, or interrupt cycles to indicate proper transfer of data between the CPU and the DUART.
INTRN	21	O	**Interrupt Request:** Active low, open drain output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
IACKN	36	I	**Interrupt Acknowledge:** Active low input indicating an interrupt acknowledge cycle. In response, the DUART will place the interrupt vector on the data bus and will assert DTACKN if it has an interrupt pending.
X1/CLK	33	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. If a crystal is used, a capacitor of approximately 15-20pF should be connected from this pin to ground.
X2	32	I	**Crystal 2:** Connection for other side of the crystal. Should be open if crystal is not used. If a crystal is used, a capacitor of approximately 15-20pF should be connected from this pin to ground.
RxDA	31	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	10	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	30	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	11	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	29	O	**Output 0.** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated automatically on receive or transmit.
OP1	12	O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated automatically on receive or transmit.
OP2	28	O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	13	O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	27	O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.

Preview

PIN DESIGNATION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
OP5	14	O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	26	O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output
OP7	15	O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output
IP0	7	I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN)
IP1	4	I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN)
IP2	37	I	**Input 2:** General purpose input, or channel B receiver external clock input (RxCB), or counter/timer external clock input. When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP3	2	I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA) When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP4	39	I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	38	I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB) When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
V_{CC}	40	I	**Power Supply:** +5V supply input.
GND	20	I	**Ground**

BLOCK DIAGRAM

The 68681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The DTACKN output is asserted during write and read cycles to indicate to the CPU that data has been latched on a write cycle, or that valid data is present on the bus on a read cycle.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurrence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR), the interrupt status register (ISR), the auxiliary control register (ACR), and the interrupt vector register (IVR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. When IACKN is asserted, and the DUART has an interrupt pending, the DUART responds by placing the contents of the IVR register on the data bus and asserting DTACKN.

Outputs OP3–OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

MARCH 1982

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC68681

Preview

Communications Channels A and B

Each communications channel of the 68681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 6-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25–50μs will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8 bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 68681 is conditioned to transmit data when the transmitter is enabled thorugh the command register. The 68681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. if it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted. If the transmitter has been disabled.

Receiver

The 68681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least sigificant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wakeup' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed when the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

Table 1. 68681 REGISTER ADDRESSING

A4	A3	A2	A1	READ (R/WN = 1)	WRITE (R/WN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	Interrupt Vector Reg. (IVR)	Interrupt Vector Reg. (IVR)
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

Table 2. REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A **MR1B**	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = special mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A **MR2B**	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0-7 if channel is programmed for 5 bits/char

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CSRA **CSRB**	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CRA **CRB**	not used—must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA **SRB**	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/FFULLB	0 = OPR[4] 1 = RxRDY/FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

Preview

Table 2. REGISTER BIT FORMATS (continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]
CTUR								

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]
CTLR								

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	IVR[7]	IVR[6]	IVR[5]	IVR[4]	IVR[3]	IVR[2]	IVR[1]	IVR[0]
IVR								

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the accumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits

Preview

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

Preview

CSRA[7:4]	Baud Rate ACR[7] = 0	Baud Rate ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4K	19.2K
1 1 0 1	Timer	Timer
1 1 1 0	IP4—16X	IP4—16X
1 1 1 1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

CSRA[3:0]	Baud Rate ACR[7] = 0	Baud Rate ACR[7] = 1
1 1 1 0	IP3—16X	IP3—16X
1 1 1 1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

CSRB[7:4]	Baud Rate ACR[7] = 0	Baud Rate ACR[7] = 1
1 1 1 0	IP2—16X	IP2—16X
1 1 1 1	IP2—1X	IP2—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

CSRB[3:0]	Baud Rate ACR[7] = 0	Baud Rate ACR[7] = 1
1 1 1 0	IP5—16X	IP5—16X
1 1 1 1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the followng:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

Preview

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

ACR[6:4]—Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change In Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR2B[5]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

Table 3. BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE:
Duty cycle of 16X clock is 50% ± 1%

Table 4. ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)[1]
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)[1]
1 0 1	Timer	External (IP2) divided by 16[1]
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

[1] In these modes, the channel B receiver clock should normally be generated from the baud rate generator.

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR2A[5]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3-OP7 or the reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSB's and eight LSB's respectively of the value to be used by the counter/timer in either the counter or timer modes of operation.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1111) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR. The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1110). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

IVR — Interrupt Vector Register

This register contains the interrupt vector. The register is initialized to H'0F' by RESET. The contents of the register are placed on the data bus when the DUART responds to a valid interrupt acknowledge cycle.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC68681

Preview

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to + 70	°C
Storage temperature	− 65 to + 150	°C
All voltages with respect to ground[3]	− 0.5 to + 6.0	V

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on + 150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient for ceramic package (116°C/W for plastic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		3.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	− 10		10	µA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	− 10		10	µA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	− 10		10	µA
I_{CC}	Power supply current				150	mA

NOTES:

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at + 25°C, typical supply voltages, and typical processing parameters.

Preview

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4, 5, 6, 7]

PARAMETER		TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset Timing (figure 1)					
t_{RES}	RESETN pulse width	0			µs
Bus Timing (figures 2, 3, 4)					
t_{AS}	A1-A4 setup time to CSN low	10			ns
t_{AH}	A1-A4 hold time from CSN low	0			ns
t_{RWS}	RWN setup time to CSN low	0			ns
t_{RWH}	RWN holdup time to CSN high	0			ns
t_{CSW}[8]	CSN high pulse width	160			ns
t_{CSD}[9]	CSN or IACKN high from DTACKN low	0			ns
t_{DD}	Data valid from CSN or IACKN low			150	ns
t_{DF}	Data bus floating from CSN or IACKN high			90	ns
t_{DS}	Data setup time to CLK high	100			ns
t_{DH}	Data hold time from CSN high	0			ns
t_{DAL}	DTACKN low from read data valid	0			ns
t_{DCR}	DTACKN low (read cycle) from CLK high			50	ns
t_{DCW}	DTACKN low (write cycle) from CLK high			30	ns
t_{DAH}	DTACKN high from CSN or IACKN high			80	ns
t_{DAT}	DTACKN high impedance from CSN or IACKN high			100	ns
t_{CSC}[10]	CSN or IACKN setup time to clock high	30			ns
Port Timing (figure 5)					
t_{PS}	Port input setup time to RDN low	0			ns
t_{PH}	Port input hold time from RDN high	0			ns
t_{PD}	Port output valid from WRN high			400	ns
Interrupt Reset Timing (figure 6)					
t_{IR}	INTRN, or OP3-OP7 when used as interrupts, high from:				
	Read RHR (RxRDY/FFULL interrupt)			300	ns
	Write THR (TxRDY interrupt)			300	ns
	Reset command (delta break interrupt)			300	ns
	Stop C/T command (counter interrupt)			300	ns
	Read IPCR (input port change interrupt)			300	ns
	Write IMR (Clear of interrupt mask bit)			200	ns
Clock Timing (figure 7)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC}	CTCLK high or low time	200			ns
f_{CTC}	CTCLK frequency	0		2.0	MHz
t_{RX}	RXC high or low time	220			ns
f_{RX}	RXC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
t_{TX}	TXC high or low time	220			ns
f_{TX}	TXC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
Transmitter Timing (figure 8)					
t_{TXD}	TXD output delay from TXC low			350	ns
t_{TCS}	TXC output skew from TXD output data	−75	0	75	ns
Receiver Timing (figure 9)					
t_{RXS}	RXD data setup time to RXC high	200			ns
t_{RXH}	RXD data hold time from RXC high	200			ns

NOTES:

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$.
8. This specification will impose maximum 68000 CPU CLK to 6MHz. Higher CPU CLK can be used if repeating bus reads are not performed.
9. This specification imposes a lower bound on CSN and IACKN low, guaranteeing that it will be low for at least 1 CLK period.
10. This specification is made only to insure that DTACKN is asserted with respect to the rising edge of the X1/CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the setup time is violated, DTACKN may be asserted as shown, or may be asserted one clock cycle later.

Preview

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to + 70	°C
Storage temperature	− 65 to + 150	°C
All voltages with respect to ground[3]	− 0.5 to + 6.0	V

NOTES

1 Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.

2 For operating at elevated temperatures, the device must be derated based on + 150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient for ceramic package (116°C/W for plastic package).

3 This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			[illegible]
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	[illegible]
V_{IH}	Input high voltage (except X1/CLK)		2.0			[illegible]
V_{IH}	Input high voltage (X1/CLK)		3.0			[illegible]
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	[illegible]
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			[illegible]
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	− 10		10	[illegible]
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	− 10		10	[illegible]
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	− 10		10	[illegible]
I_{CC}	Power supply current				150	[illegible]

NOTES

4 Parameters are valid over specified temperature range.

5 All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.

6 Typical values are at + 25°C, typical supply voltages, and typical processing parameters.

MARCH 1982

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SC68681

Preview

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4, 5, 6, 7]

PARAMETER		TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset Timing (figure 1)					
t_{RES}	RESETN pulse width	0			μs
Bus Timing (figures 2, 3, 4)					
t_{AS}	A1-A4 setup time to CSN low	10			ns
t_{AH}	A1-A4 hold time from CSN low	0			ns
t_{RWS}	RWN setup time to CSN low	0			ns
t_{RWH}	RWN holdup time to CSN high	0			ns
t_{CSW}[8]	CSN high pulse width	160			ns
t_{CSD}[9]	CSN or IACKN high from DTACKN low	0			ns
t_{DD}	Data valid from CSN or IACKN low			150	ns
t_{DF}	Data bus floating from CSN or IACKN high			90	ns
t_{DS}	Data setup time to CLK high	100			ns
t_{DH}	Data hold time from CSN high	0			ns
t_{DAL}	DTACKN low from read data valid	0			ns
t_{DCR}	DTACKN low (read cycle) from CLK high			50	ns
t_{DCW}	DTACKN low (write cycle) from CLK high			30	ns
t_{DAH}	DTACKN high from CSN or IACKN high			80	ns
t_{DAT}	DTACKN high impedance from CSN or IACKN high			100	ns
t_{CSC}[10]	CSN or IACKN setup time to clock high	30			ns
Port Timing (figure 5)					
t_{PS}	Port input setup time to RDN low	0			ns
t_{PH}	Port input hold time from RDN high	0			ns
t_{PD}	Port output valid from WRN high			400	ns
Interrupt Reset Timing (figure 6)					
t_{IR}	INTRN, or OP3-OP7 when used as interrupts, high from:				
	Read RHR (RxRDY/FFULL interrupt)			300	ns
	Write THR (TxRDY interrupt)			300	ns
	Reset command (delta break interrupt)			300	ns
	Stop C/T command (counter interrupt)			300	ns
	Read IPCR (input port change interrupt)			300	ns
	Write IMR (Clear of interrupt mask bit)			200	ns
Clock Timing (figure 7)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC}	CTCLK high or low time	200			ns
f_{CTC}	CTCLK frequency	0		2.0	MHz
t_{RX}	RXC high or low time	220			ns
f_{RX}	RXC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
t_{TX}	TXC high or low time	220			ns
f_{TX}	TXC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
Transmitter Timing (figure 8)					
t_{TXD}	TXD output delay from TXC low			350	ns
t_{TCS}	TXC output skew from TXD output data	−75	0	75	ns
Receiver Timing (figure 9)					
t_{RXS}	RXD data setup time to RXC high	200			ns
t_{RXH}	RXD data hold time from RXC high	200			ns

NOTES

[4] Parameters are valid over specified temperature range.

[5] All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.

[6] Typical values are at +25°C, typical supply voltages, and typical processing parameters.

[7] Test condition for outputs: $C_L = 150pF$.

[8] This specification will impose maximum 68000 CPU CLK to 6MHz. Higher CPU CLK can be used if repeating bus reads are not performed.

[9] This specification imposes a lower bound on CSN and IACKN low, guaranteeing that it will be low for at least 1 CLK period.

[10] This specification is made only to insure that DTACKN is asserted with respect to the rising edge of the X1/CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the setup time is violated, DTACKN may be asserted as shown, or may be asserted one clock cycle later.

Preview

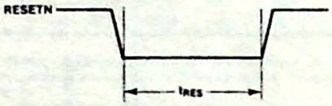

RESETN
tRES

Figure 1. Reset Timing

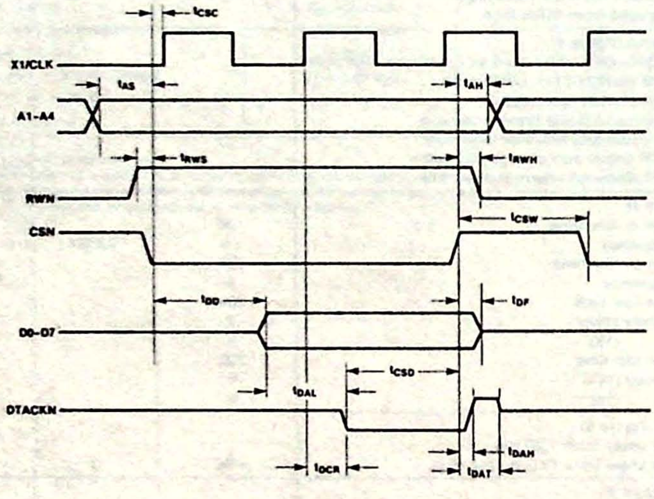

tCSC
X1/CLK
tAS
tAH
A1-A4
tRWS
tRWH
RWN
CSN
tCSW
tDD
tDF
D0-D7
tCSD
tDAL
DTACKN
tDAH
tDCR
tDAT

Figure 2. Bus Timing (Read Cycle)

Preview

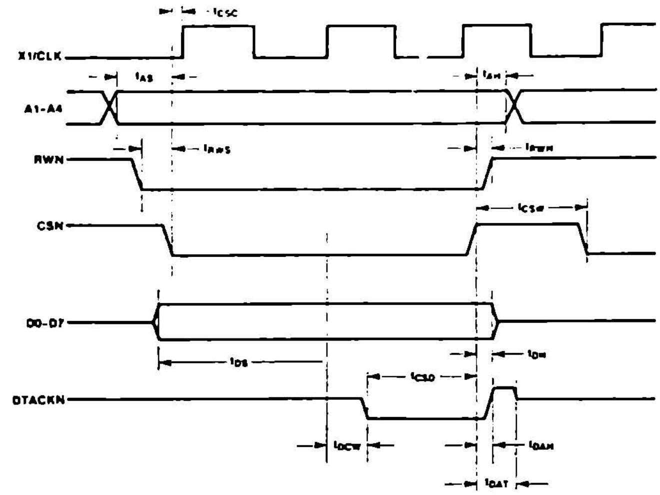

tCSC
X1/CLK
tAS
tAH
A1-A4
RWN
tRWS
tRWH
CSN
tCSW
D0-D7
tDS
tDH
tCSD
DTACKN
tDCW
tDAH
tDAT

Figure 3. Bus Timing (Write Cycle)

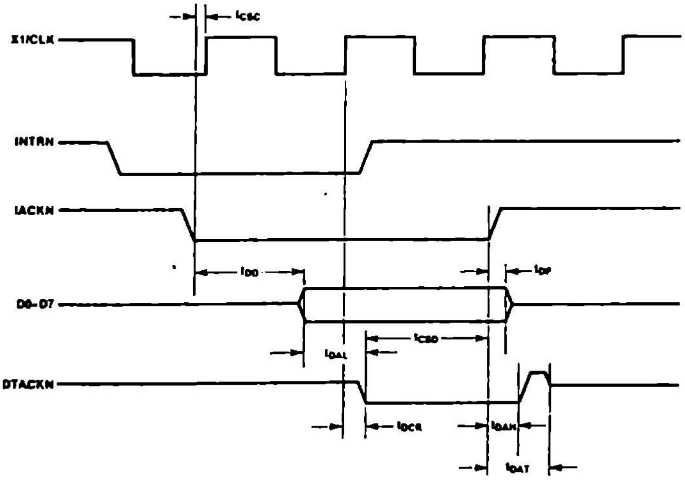

tCSC
X1/CLK
INTRN
IACKN
tDD
tDF
D0-D7
tCSD
tDAL
DTACKN
tDCR
tDAH
tDAT

Figure 4. Interrupt Cycle Timing

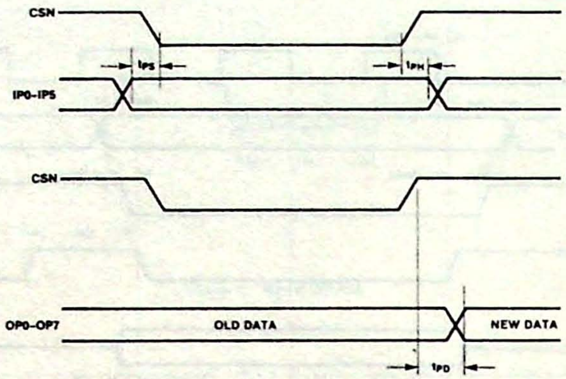
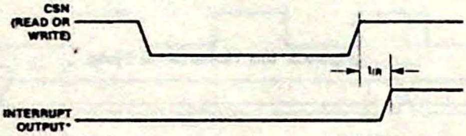

CSN
t_{PS}
t_{PH}
IP0–IP5
CSN
OP0–OP7
OLD DATA
NEW DATA
t_{PD}

Figure 5. Port Timing


CSN (READ OR WRITE)
t_{IR}
INTERRUPT OUTPUT*

*INTRN or OP3–OP7 when used as interrupt outputs.

Figure 6. Interrupt Timing

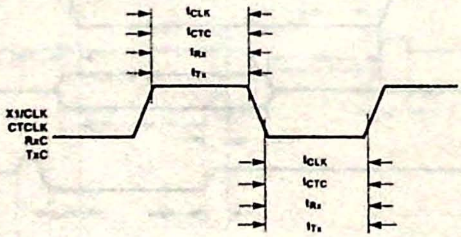

t_{CLK}
t_{CTC}
t_{Rx}
t_{Tx}
X1/CLK
CTCLK
RxC
TxC
t_{CLK}
t_{CTC}
t_{Rx}
t_{Tx}

Figure 7. Clock Timing

Preview

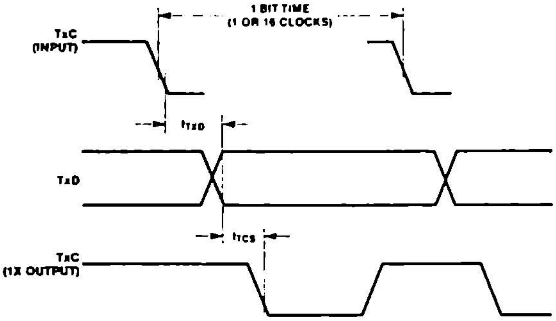

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
t_{TXD}
TxD
t_{TCS}
TxC
(1X OUTPUT)

Figure 8. Transmit Timing

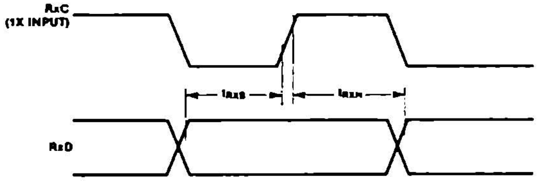
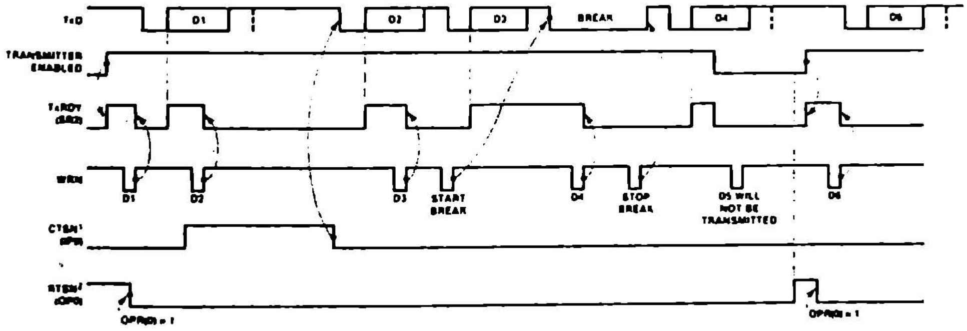

RxC
(1X INPUT)
t_{RXS}
t_{RXH}
RxD

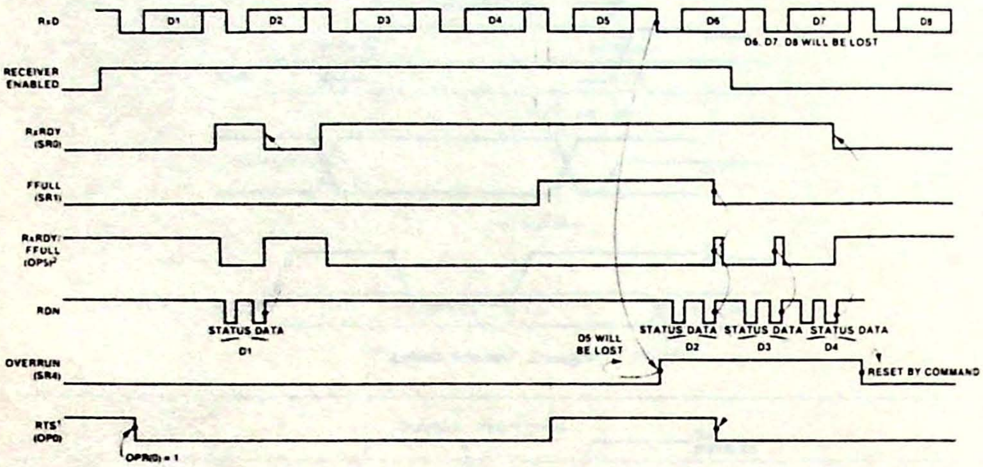
Figure 9. Receive Timing


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
WRN
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN¹
(IP0)
RTSN²
(OP0)
OPR(0) = 1
OPR(0) = 1

NOTES
1 TIMING SHOWN FOR MR2(4) = 1
2 TIMING SHOWN FOR MR2(5) = 1

Figure 10. Transmitter Timing

Preview


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFULL (SR1)
RxRDY/FFULL (OP5)²
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS¹ (OP0)
OPR(0) = 1

NOTES
1. TIMING SHOWN FOR MR1(7) = 1.
2. SHOWN FOR OPCR(4) = 1 AND MR1(5) = 0.

Figure 11. Receiver Timing

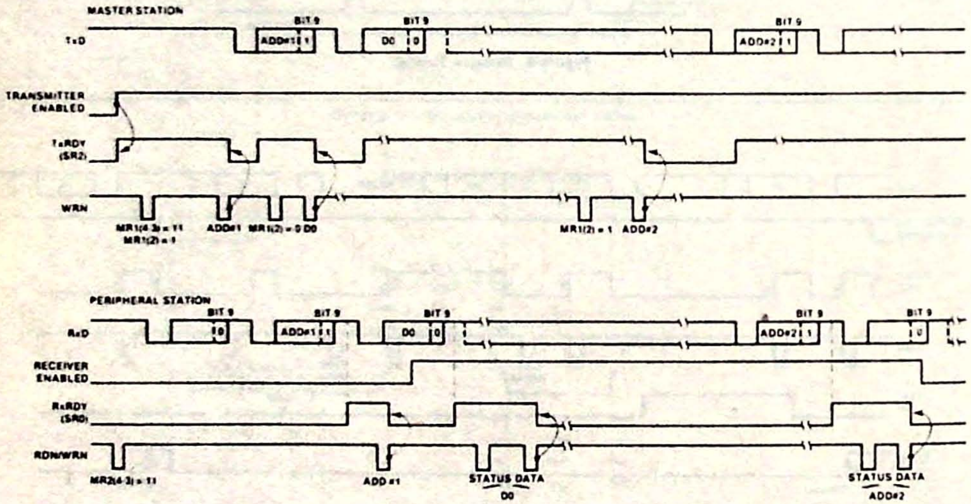
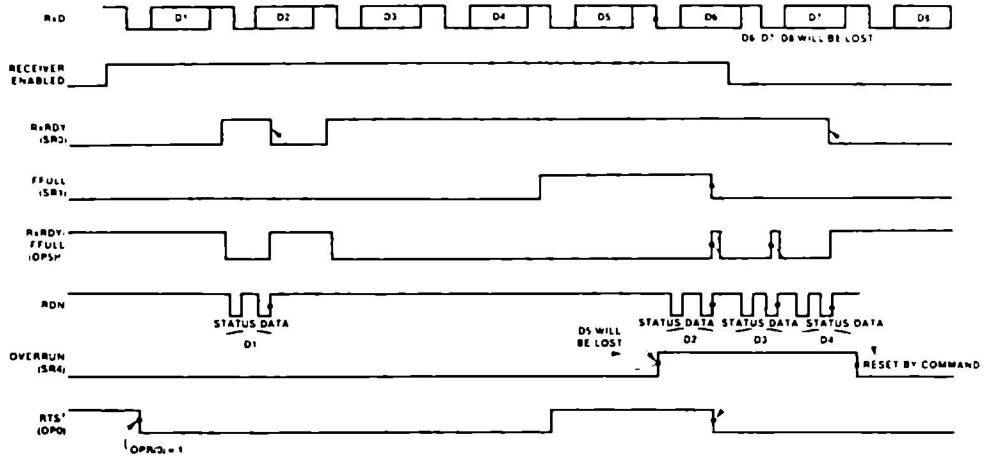

MASTER STATION
TxD
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1(4-3) = 11
MR1(2) = 1
ADD#1
MR1(2) = 0
D0
MR1(2) = 1
ADD#2
PERIPHERAL STATION
RxD
BIT 9
0
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
BIT 9
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR2(4-3) = 11
ADD #1
STATUS DATA
D0
STATUS DATA
ADD#2

Figure 12. Wake up Mode

Section 5
Video Game

VALVO


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6 D7 D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFULL (SR1)
RxRDY/FFULL (OP5)
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS' (OP0)
OPR(0) = 1

NOTES
1 TIMING SHOWN FOR MR1(7) = 1
2 SHOWN FOR OPCR(4) = 1 AND MR1(6) = 0

Figure 11. Receiver Timing

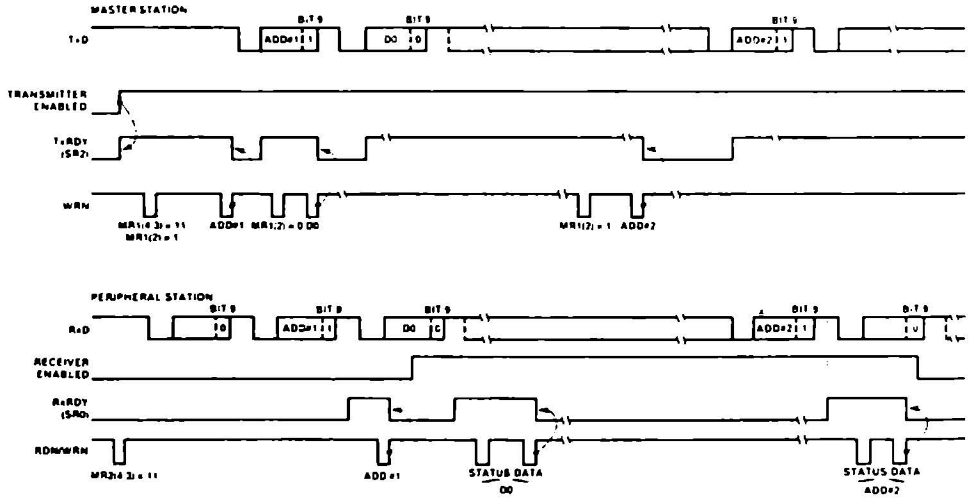

MASTER STATION
TxD
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1(4 3) = 11
MR1(2) = 1
ADD#1
MR1(2) = 0 D0
MR1(2) = 1
ADD#2
PERIPHERAL STATION
RxD
BIT 9
0
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
BIT 9
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR1(4 3) = 11
ADD #1
STATUS DATA
D0
STATUS DATA
ADD#2

Figure 12. Wake up Mode

Section 5
Video Game

VALVO

JANUARY 1982

MICROPROCESSOR MAB 2650 A

PRODUCT BRIEF

DESCRIPTION

The Signetics 2650A series are 8-bit general purpose microprocessors constructed using Signetics n-channel silicon gate MOS technology. The 2650 series executes a fixed instruction set, with each instruction being one to three bytes in length.

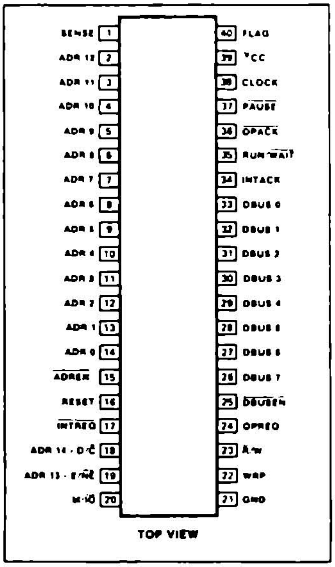
The 2650 instruction set consists of many powerful instructions which are all easily understood and are typical of larger computers. There are one-, two-, and three-byte instructions as a result of the multiplicity of addressing modes.

Addressing range of these processors is 32K bytes of memory and 256 I/O devices. A single level hardware vectored interrupt capability is provided.

FEATURES

- Static 8 bit parallel NMOS microprocessor
- Single power supply of +5 volts
- TTL level single phase clock
- TLL compatible inputs and outputs
- Variable length instructions of 1, 2 or 3 bytes
- 32K byte addressing range
- Coding efficiency with multiple addressing modes
- Synchronous or asynchronous memory and I/O interface
- Interfaces directly with industry standard memories
- Single bit serial I/O path
- Seven 8 bit addressable general purpose registers
- Vectored interrupt
- Subroutine return address stack

PIN CONFIGURATION


SENSE 1
ADR 12 2
ADR 11 3
ADR 10 4
ADR 9 5
ADR 8 6
ADR 7 7
ADR 6 8
ADR 5 9
ADR 4 10
ADR 3 11
ADR 2 12
ADR 1 13
ADR 0 14
ADREN 15
RESET 16
INTREQ 17
ADR 14 - D/C 18
ADR 13 - E/NE 19
M/IO 20
40 FLAG
39 VCC
38 CLOCK
37 PAUSE
36 OPACK
35 RUN/WAIT
34 INTACK
33 DBUS 0
32 DBUS 1
31 DBUS 2
30 DBUS 3
29 DBUS 4
28 DBUS 5
27 DBUS 6
26 DBUS 7
25 DBUSEN
24 OPREQ
23 R/W
22 WRP
21 GND
TOP VIEW

MICROPROCESSOR BLOCK DIAGRAM

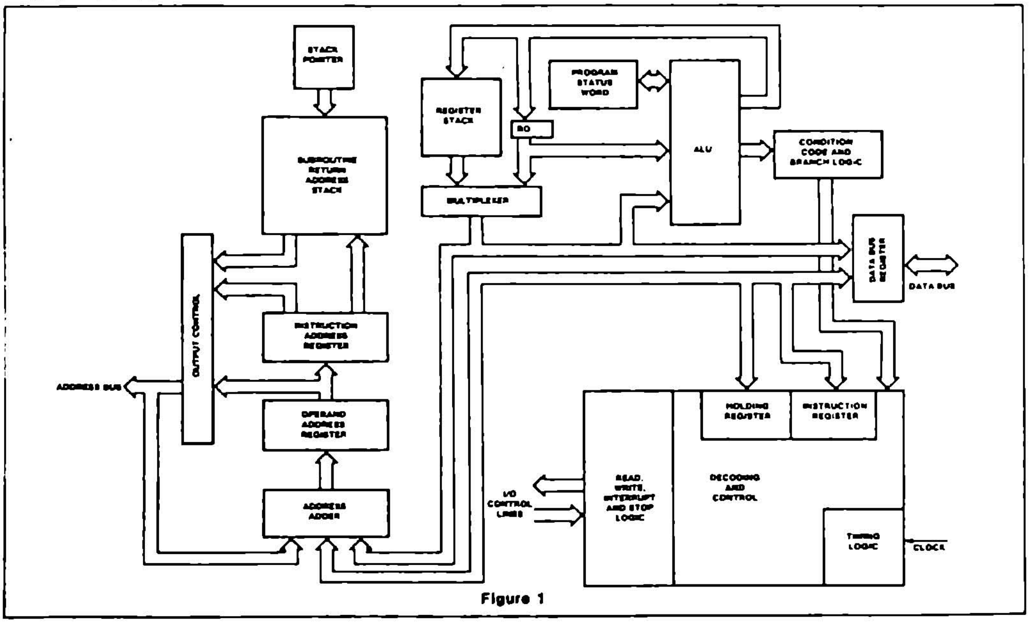

STACK POINTER
SUBROUTINE RETURN ADDRESS STACK
OUTPUT CONTROL
INSTRUCTION ADDRESS REGISTER
ADDRESS BUS
OPERAND ADDRESS REGISTER
ADDRESS ADDER
REGISTER STACK
R0
MULTIPLEXER
PROGRAM STATUS WORD
ALU
CONDITION CODE AND BRANCH LOGIC
DATA BUS REGISTER
DATA BUS
HOLDING REGISTER
INSTRUCTION REGISTER
I/O CONTROL LINES
READ, WRITE, INTERRUPT AND STOP LOGIC
DECODING AND CONTROL
TIMING LOGIC
CLOCK

Figure 1

PIN DESIGNATION

MNEMONIC	NUMBER	NAME	TYPE	FUNCTION
ADR0-ADR12	14-2	Address lines	O	Low order memory address lines for instruction or operand fetch. ADR0 is the least significant bit and ADR12 is the most significant bit. ADR0 through ADR7 are also used as the I/O device address for extended I/O instructions.
ADR13-E/$\overline{\text{NE}}$	19	Address 13-Extended/Non extended	O	Low order memory page address line during memory reference instructions. For I/O instructions this line discriminates between extended and non-extended I/O instructions.
ADR14-D/$\overline{\text{C}}$	18	Address 14-Data/Control	O	High order memory page address line during memory reference instructions. It also serves as the I/O device address for non-extended I/O instructions.
$\overline{\text{ADREN}}$	15	Address enable	I	Active low input allowing 3-state control of the address bus ADR0-ADR12.
DBUS0-DBUS7	33-26	Data bus	I/O	These lines provide communication between the CPU, Memory, and I/O devices for instruction and data transfers.
$\overline{\text{DBUSEN}}$	25	Data bus enable	I	This active low input allows tri-state control of the data bus.
OPREQ	24	Operation request	O	Indicates to external devices that all address, data and control information is valid.
$\overline{\text{OPACK}}$	36	Operation acknowledge	I	Active low input indicating completion of an external operation. This allows asynchronous functioning of external devices.
M/$\overline{\text{IO}}$	20	Memory/Input-output	O	Indicates whether the current operation references memory or I/O.
$\overline{\text{R}}$/W	23	Read/Write	O	Indicates a read or a write operation.
WRP	22	Write pulse	O	This is a timing signal from the 2650 that provides a positive-going pulse during each requested write operation (memory or I/O) and a high level during read operations.
SENSE	1	Sense	I	The sense bit in the PSU reflects the logic state of the sense input to the processor at pin #1.
FLAG	40	Flag	O	The flag bit in the PSU is tied to a latch that drives the flag output at pin #40.
$\overline{\text{INTREQ}}$	17	Interrupt request	I	This active low input line indicates to the processor that an external device is requesting service. The processor will recognize this signal at the end of the current instruction if the interrupt inhibit status bit is zero.
INTACK	34	Interrupt acknowledge	O	This line indicates that the 2650 is ready to receive the interrupt vector (relative address byte) from the interrupting device.
$\overline{\text{PAUSE}}$	37	Pause	I	This active low input is used to suspend processor operation at the end of the current instruction.
RUN/$\overline{\text{WAIT}}$	35	Run/Wait	O	This output is a processor status indicator. During normal operation this line is high. If the processor is halted either by executing a halt instruction or by a low input on the pause line, the run/wait line will go low.
RESET	16	Reset	I	Resets the instruction address register to zero. Clears interrupt inhibit.
CLOCK	38	Clock	I	A positive going pulse train that determines the instruction execution time.
V_{CC}	39	+5V supply	I	+5V power
GND	21	Ground	I	Ground

FUNCTIONAL DESCRIPTION

The 2650 series processors are general purpose, single chip, fixed instruction set, parallel 8-bit binary processors. A general purpose processor can perform any data manipulations through execution of a stored sequence of machine instructions. The processor has been designed to closely resemble conventional binary computers, but executes variable length instructions of one to three bytes in length.

The 2650 series contains a total of seven general purpose registers, each eight bits long. They may be used as source or destination for arithmetic operations, as index registers, and for I/O transfers.

The processor can address up to 32,768 bytes of memory in four pages of 8,192 bytes each. The processor instructions are one, two, or three bytes long, depending on the instruction. Variable length instructions tend to conserve memory space since a one-or-two byte instruction may often be used rather than a three byte instruction. The first byte of each instruction always specifies the operation to be performed and the addressing mode to be used. Most instructions use six of the first eight bits for this purpose, with the remaining two bits forming the register field. Some instructions use the full eight bits as an operation code.

The data bus and address signals are tri-state to provide convenience in system design. Memory and I/O interface signals are asynchronous so that*direct memory access (DMA) and multiprocessor operations are easy to implement.

The block diagram for the 2650 series (figure 1) shows the major internal components and the data paths that interconnect them. In order for the processor to execute an instruction, it performs the following general steps:

1. The instruction address register provides an address for memory.
2. The first byte of an instruction is fetched from memory and stored in the instruction register.
3. The instruction register (IR) is decoded to determine the type of instruction and the addressing mode.
4. If an operand from memory is required, the operand address is resolved and loaded into the operand address register.
5. The operand is fetched from memory and the operation is executed.
6. The first byte of the next instruction is fetched.

*The instruction register holds the first byte of each instruction and directs the subsequent operations required to execute each instruction. The IR contents are decoded and used in conjunction with the timing information to control the activation and sequencing of all the other elements on the chip. The holding register is used in some multiple-byte instructions to contain further instruction information and partial absolute addresses.

The arithmetic logic unit (ALU) is used to perform all of the data manipulation operations, including load, store, add, subtract, AND, inclusive OR, exclusive OR, compare, rotate, increment and decrement. It contains and controls the carry bit, the overflow bit, the interdigit carry and the condition code register.

The register stack contains six registers that are organized into two banks of three registers each. The register select bit picks one of the two banks to be accessed by instructions. In order to accommodate the register-to register instructions, register zero (R0) is outside the array. Thus, register zero is always available along with one set of three registers.

The address adder is used to increment the instruction address and to calculate relative and indexed addresses.

The instruction address register holds the address of the next instruction byte to be accessed. The operand address register stores operand addresses and sometimes contains intermediate results during effective address calculations.

The return address stack (RAS) is a last in, first out (LIFO) storage which receives the return address whenever a branch-to-subroutine instruction is executed. When a return instruction is executed, the RAS provides the last return address for the processor's IAR. The stack contains eight levels of storage so that subroutines may be nested up to eight levels deep. The stack pointer is a three bit wraparound counter that indicates the next available level in the stack. It always points to the current address.

PROGRAM STATUS WORD

The program status word (PSW) is a major feature of the 2650 which greatly increases its flexibility and processing power. The PSW is a special purpose register within the processor that contains status and control bits.

It is divided into two bytes called the program status upper (PSU) and program status lower (PSL). The PSW bits may be tested, loaded, stored, preset, or cleared using the instructions which affect the PSW. The bits are utilized as shown in table 1.

Table 1 PROGRAM STATUS WORD

PSU0,1,2	SP	Pointer for the return address stack
PSU3,4		Not used. These bits are always zero.
PSU5	II	Used to inhibit recognition of additional interrupts.
PSU6	F	Flag is a latch directly driving the flag output.
PSU7	S	Sense equals the state of the sense input
PSL0	C	Carry stores any carry from the high-order bit of ALU
PSL1	COM	Compare determines if a logical or arithmetic comparison is to be made.
PSL2	OVF	Overflow is set if a two's complement overflow occurs.
PSL3	WC	With carry determines if the carry is used in arithmetic and rotate instructions.
PSL4	RS	Register select identifies which bank of 3 GP registers is being used.
PSL5	IDC	Inter digit carry stores the bit-3 to bit-4 carry in arithmetic operations.
PSL6,7	CC	Condition code is affected by compare, test and arithmetic instructions.

PSU

7	6	5	4	3	2	1	0
S	F	II	—	—	SP2	SP1	SP0

S	Sense
F	Flag
II	Interrupt inhibit
SP2	Stack pointer two
SP1	Stack pointer one
SP0	Stack pointer zero

PSL

7	6	5	4	3	2	1	0
CC1	CC0	IDC	RS	WC	OVF	COM	C

CC1	Condition code one
CC0	Condition code zero
IDC	Interdigit carry
RS	Register bank select
WC	With/without carry
OVF	Overflow
COM	Logical arithmetic compare
C	Carry/borrow

INPUT/OUTPUT INTERFACE

The 2650 series microprocessor has a set of versatile I/O instructions and can perform I/O operations in a variety of ways. One-and two byte I/O instructions are provided, as well as a special single-bit I/O facility. The I/O modes provided by the 2650 are designated as data, control, and extended I/O

Data or control I/O instructions, also called non-extended I/O instructions, are one byte long. Any general purpose register can be used as the source or destination. A special control line indicates if either a data or control instruction is being executed.

Extended I/O is a two-byte read or write instruction. Execution of an extended I/O instruction will cause an 8-bit address, taken from the second byte of the instruction, to be placed on the low order eight address lines. The data, which can originate or terminate with any general purpose register, is placed on the data bus. This type of I/O can be used to simultaneously select a device and send data to it.

Memory reference instructions that address data outside of physical memory may also be used for I/O operations. When an instruction is executed, the address may be decoded by the I/O device rather than memory.

MEMORY INTERFACE

The memory interface consists of the address bus, the 8-bit data bus and several signals that operate in an interlocked or handshaking mode.

The write pulse signal is designed to be used as a memory strobe signal for any memory type. It has been particularly optimized to be used as the chip enable or read/write signal.

INTERRUPT HANDLING CAPABILITY

The 2650 series has a single level hardware vectored interrupt capability. When an interrupt occurs, the processor finishes the current instruction and sets the interrupt inhibit bit in the PSW. The processor then executes a branch to subroutine relative to location zero (ZBSR) instruction and sends out interrupt acknowledge and operation request signals. On receipt of the INTACK signal, the interrupting device inputs an 8-bit address, the interrupt vector, on the data bus. The relative and relative indirect addressing modes combined with this 8-bit address allow interrupt service routines to begin at any addressable memory location.

INSTRUCTION SET

The 2650 instruction set consists of many powerful instructions which are all easily understood and are typical of larger computers. There are one-, two-, and three-byte instructions as a result of the multiplicity of addressing modes.

Automatic incrementing or decrementing of an index register is available in the arithmetic indexed instructions. All of the branch instructions except indexed branching can be conditional.

Register-to-register instructions are one byte; register-to-storage instructions are two or three bytes long. The two-byte register-to-memory instructions are either immediate or relative addressing types.

JANUARY 1982

UNIVERSAL SYNC GENERATORS (USG) MEB 2621 (PAL), MEB 2622 (NTSC)

PRODUCT BRIEF

DESCRIPTION

The Signetics 2621 Universal Sync Generator (USG) provides the timing and control signals necessary for generating and displaying TV video information in the PAL format.

The USG accepts a single 3.55MHz input clock and generates various timing outputs including vertical, horizontal and composite blanking, composite sync and color burst flag. Several auxiliary clock outputs are also provided.

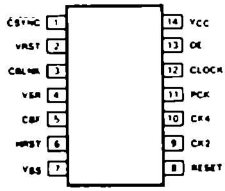
The USG is primarily intended for use in microprocessor-controlled video games. A typical game configuration consists of a 2621 USG, a 2650A microprocessor, a 2636 Programmable Video Interface, a 2616 16K ROM, and digital video summer circuitry.

The 2621 is constructed using Signetics silicon gate N-channel depletion load technology and operates from a single +5 volt power supply.

The Signetics 2622 Universal Sync Generator (USG) provides the timing and control signals necessary for generating and displaying TV video information in the NTSC format.

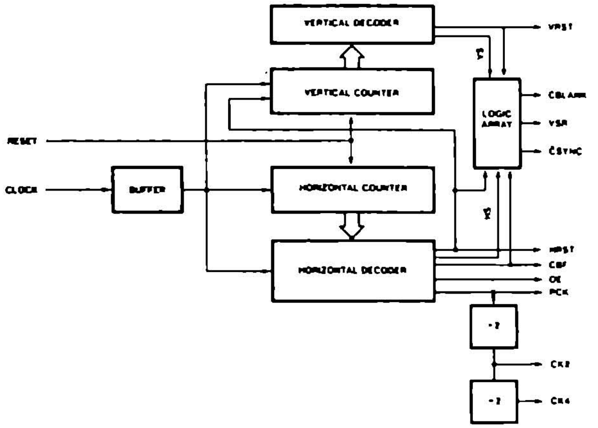
The USG accepts a single 3.5795MHz input clock and generates various timing outputs including vertical, horizontal, and composite blanking, composite sync and color burst flag. Several auxiliary clock outputs are also provided. The USG is primarily intended for use in microprocessor-controlled video games. A typical game configuration consists of a 2622 USG, a 2650A microprocessor, a 2636 Programmable Video Interface, a 2616 16K ROM, and video summer circuitry. The 2622 is constructed using Signetics silicon gate N-channel depletion load technology and operates from a single +5 volt power supply.

PIN CONFIGURATION


C̄SYNC 1
VRST 2
CBLANK 3
VSR 4
CBF 5
HRST 6
VSS 7
14 VCC
13 OE
12 CLOCK
11 PCK
10 CK4
9 CK2
8 RESET

TOP VIEW

BLOCK DIAGRAM


VERTICAL DECODER
VERTICAL COUNTER
RESET
CLOCK
BUFFER
HORIZONTAL COUNTER
HORIZONTAL DECODER
LOGIC ARRAY
VRST
CBLANK
VSR
C̄SYNC
HRST
CBF
OE
PCK
÷2
CK2
÷2
CK4

JANUARY 1982

PROGRAMMABLE VIDEO INTERFACE (PVI) — MEB 2636

PRODUCT BRIEF

DESCRIPTION

The Signetics 2636 Programmable Video Interface (PVI) is intended for use in microprocessor-controlled game systems, and provides all of the common game circuits on a single chip. Circuits are provided for player inputs, background, moving objects, scoring, and audio signals.

A typical system configuration consists of five LSI circuits: a PVI, a 2616 16K ROM, an NE549 Digital Video Summer (DVS) a Universal Sync Generator (USG), and a 2650A microprocessor.

Additional PVIs as well as random logic can easily be interfaced to enhance game complexity. Since the system is microprocessor based, the actual game itself need not be "hardwired" into the system. Game definition is completely contained in the ROM. To change games, one simply replaces one ROM with another. Each ROM can contain several games, depending on game complexity and similarity between games.

The 2636 PVI is constructed using Signetics' silicon gate N-Channel depletion load technology and operates from a single +5 volt power supply

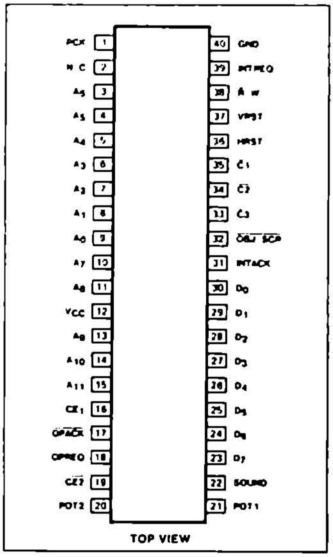
FEATURES

- Four general-purpose, RAM-resident object modules
- Object duplication permitting generation of up to 80 object images on the screen
- 280ns object resolution
- Object size and position under program control
- Programmable score
- Programmable sound
- Programmable background
- Eight programmable colors with multiple brightness levels
- 37-byte scratch pad memory
- Chip Enable outputs for system ROMs and PROMs
- I/O facilities for switch scanning and potentiometer inputs
- Wire-OR expansion capability to multiple PVIs
- Forty-pin dual-in-line package

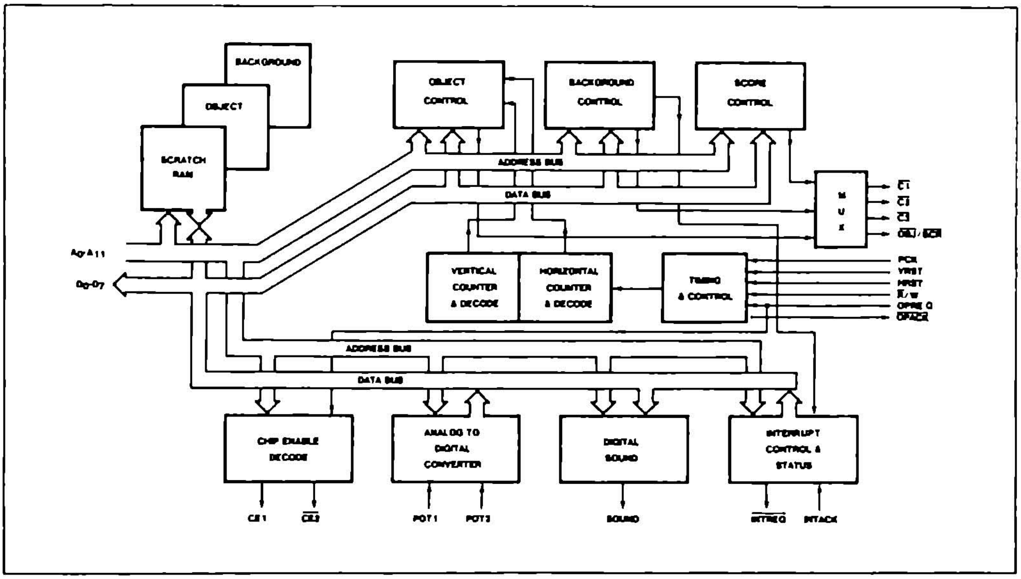
APPLICATIONS

- Consumer programmable video games
- Arcade games
- Simulators
- Special purpose graphic displays
- Home computer center

PIN CONFIGURATION


PCX 1
N C 2
A6 3
A5 4
A4 5
A3 6
A2 7
A1 8
A0 9
A7 10
A8 11
VCC 12
A9 13
A10 14
A11 15
CE1 16
OPACK 17
OPREQ 18
CE2 19
POT2 20
40 GND
39 INTREQ
38 R/W
37 VRST
36 HRST
35 C1
34 C2
33 C3
32 OBJ/SCR
31 INTACK
30 D0
29 D1
28 D2
27 D3
26 D4
25 D5
24 D6
23 D7
22 SOUND
21 POT1
TOP VIEW

PVI BLOCK DIAGRAM


BACKGROUND
OBJECT
SCRATCH RAM
OBJECT CONTROL
BACKGROUND CONTROL
SCORE CONTROL
ADDRESS BUS
DATA BUS
A0-A11
D0-D7
M U X
C1
C2
C3
OBJ/SCR
VERTICAL COUNTER & DECODE
HORIZONTAL COUNTER & DECODE
TIMING & CONTROL
PCK
VRST
HRST
R/W
OPREQ
OPACK
ADDRESS BUS
DATA BUS
CHIP ENABLE DECODE
ANALOG TO DIGITAL CONVERTER
DIGITAL SOUND
INTERRUPT CONTROL & STATUS
CE1
CE2
POT1
POT2
SOUND
INTREQ
INTACK

Section 6
Bipolar 8-Bit Microprocessor-Family

VALVO

FEATURES

- MINIMUM PARTS COUNT
- BIPOLAR DEVICE USING LOW POWER SCHOTTKY TECHNOLOGY
- HIGH PERFORMANCE
- SOURCE/DESTINATION ARCHITECTURE
- DESIGN FLEXIBILITY
- AFTER-SALES SUPPORT

USER BENEFITS

- Reduced Cost
- Extended Product Life and Reliability

- Full System Drive Without Buffering
- TTL Compatibility
- Proven Reliability

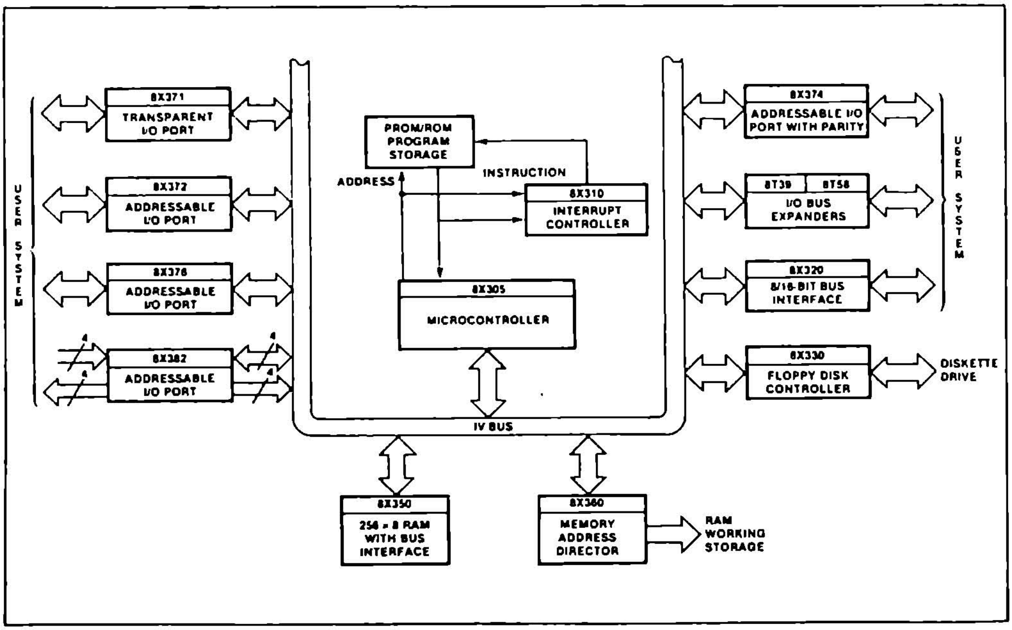
- Instruction Cycle Time of 200ns
- Bit Addressability — Instruction Data Formats of 1-to-8 Bits Without Added Delay
- Bit-Slice Capabilities at Bipolar Speeds

- Efficient Use of Fixed Instruction Set
- Easy to Program

- Single-Chip Processor
- Full Complement of Support Devices
- Second-Source Supplier

- Cross Assembler Program
- Development System
- Training Tapes, Seminars
- Complete Documentation
- Applications Support from Qualified Field Engineers

8X300 FAMILY


USER SYSTEM
8X371
TRANSPARENT I/O PORT
8X372
ADDRESSABLE I/O PORT
8X376
ADDRESSABLE I/O PORT
8X382
ADDRESSABLE I/O PORT
4
4
4
4
PROM/ROM PROGRAM STORAGE
ADDRESS
INSTRUCTION
8X310
INTERRUPT CONTROLLER
8X305
MICROCONTROLLER
IV BUS
8X374
ADDRESSABLE I/O PORT WITH PARITY
8T39
8T58
I/O BUS EXPANDERS
8X320
8/16-BIT BUS INTERFACE
USER SYSTEM
8X330
FLOPPY DISK CONTROLLER
DISKETTE DRIVE
8X350
256 × 8 RAM WITH BUS INTERFACE
8X360
MEMORY ADDRESS DIRECTOR
RAM WORKING STORAGE

PRODUCT LINE OVERVIEW

The Bipolar LSI Microprocessor product line is centered around the high-speed, Schottky fabricated 8X305 MicroController and a large family of I/O devices, support ICs, and development support tools to expedite and simplify system design. No other microprocessor product line offers the advantages of the 8X300 Family in systems requiring intelligent control of high-speed data streams.

The 8X305 MicroController can fetch, decode and execute each instruction in one 200ns machine cycle. With a single instruction, 8 bits of data can be read from the bus, latched, rotated, masked, combined with other data in an ALU operation, shifted, merged with original bus data, and returned to the bus. The architecture and instruction set of the processor is designed to provide high data throughput with both bit and byte oriented operations. The 8X305 offers more on-chip registers and significant data handling capability improvements over its predecessor, the 8X300.

Because of its extremely high speed, the 8X305 MicroController is able to perform through software many operations that would otherwise require additional system hardware. For example, using the MicroController, the 8X330 Floppy Disk Controller, and other support devices, a complete diskette drive controller can be built using only 10 integrated circuits. The resulting controller is programmable and capable of supporting multiple drive types and formats. Low parts counts typical of 8X305 based designs result in reduced assembly and testing time, lower power consumption and improved reliability.

The 8X300 Family is implemented using bipolar low-power Schottky technology to provide TTL speeds and drive capability without additional buffering. In addition, the family is compatible with most special purpose devices often required in control applications. The excellent environmental characteristics of the technology make the 8X300 Family ideal for military applications as well.

A complete complement of development support tools are offered to simplify design using the 8X305. A self-contained universal development system is available that supports full speed in-circuit emulation. Software tools are provided to enable use of mainframe or minicomputers to generate software. Complete documentation is in place, including both data sheets and comprehensive reference manuals. In addition, training aids, evaluation tools, and demonstration systems are available. To supplement these tools, Signetics employs a large professional staff of application engineers both in the field and at the factory to support 8X300 Family design efforts.

8X300 FAMILY REFERENCE TABLE

PART NO.	PRODUCT DESCRIPTION	FUNCTION
8X300	Micro Controller	250ns processor for I/O and data control
8X305	Micro Controller	200ns processor for I/O and data control
8X310	Interrupt Controller	3 prioritized interrupts with 4-level stack
8X320	Bus Interface Array	2-port RAM for 8/16-bit mailbox interfacing
8X330	Floppy Disk Controller	1 Mb/sec data rate, programmable, ECC support
8X350	Bipolar RAM	256x8 high-speed memory with bus interface
8X353	Bipolar RAM	32x8 high-speed memory with bus interface
8X355	LIFO Memory	32x8 high-speed LIFO stack with bus interface
8X360	Memory Address Director	16-bit address controller for working storage
8X371	Transparent I/O Port	8-bit bidirectional
8X372	Addressable I/O Port	8-bit bidirectional, synchronous
8X374	Addressable I/O Port	8-bit bidirectional, synchronous, with parity
8X376	Addressable I/O Port	8-bit bidirectional, asynchronous
8X382	Addressable I/O Port	4-in / 4-out
8T31	Transparent I/O Port	8-bit bidirectional
8X31 +)	Transparent I/O Port	8-bit bidirectional
8T32	Addressable I/O Port	8-bit bidirectional, synchronous
8X32 +)	Addressable I/O Port	8-bit bidirectional, synchronous
8T33 +)	Addressable I/O Port	8-bit bidirectional, synchronous
8T35 +)	Addressable I/O Port	8-bit bidirectional, asynchronous
8T36	Addressable I/O Port	8-bit bidirectional, asynchronous
8X36 +)	Addressable I/O Port	8-bit bidirectional, asynchronous
8T39 +)	Addressable Bus Expander	High-drive bus driver
8X42 +)	Addressable I/O Port	4-in / 4-out
8T58 +)	Transparent Bus Expander	High-drive bus driver
82HS137	Bipolar PROM	4.096 bit (1024x4), T.S.
82HS147	Bipolar PROM	4.096 bit (512x8), T.S.
82HS181	Bipolar PROM	8.192 bit (1024x8), T.S.
82HS185	Bipolar PROM	8.192 bit (2048x4), T.S.
82HS191	Bipolar PROM	16.384 bit (2048x8), T.S.
82HS195	Bipolar PROM	16.384 bit (4096x4), T.S.

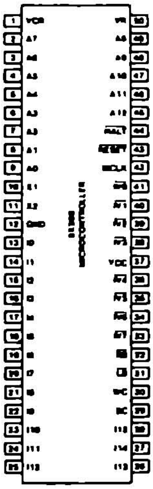
+) not for new development

ARCHITECTURAL OVERVIEW

The Signetics 8X300 Microcontroller (Figure 1) is a high-speed bipolar microprocessor implemented with low-power Schottky technology. The 8X300 brings together all the qualities needed—SPEED, FLEXIBILITY, and ECONOMY—for systems design in the many areas that require reliable bit stream management. Consider!—5V operation, TTL bus compatibility, and an on-chip clock—the result, a system with fewer parts. Consider!—the inherent power of LSI logic (programmable Rotate, Mask, Shift, and Merge functions in the data-processing path) and the ability to Fetch, Decode, and Execute a 16-bit instruction in a minimum of 250-nanoseconds—the result, a system with superior bit handling capabilities. Consider!—the 250ns cycle time in conjunction with extended microcode—the result, the flexibility of bit-slice devices with the programming ease of MOS microprocessors. Now, consider the results!—a device tailored to bit-stream management in the areas of Industrial Control, Input/Output Control, and Data Communications.

The 8X300 uses three separate buses—one for 13-bit instruction addresses, one for 16-bit instructions, and a bidirectional 8-bit input/output data bus; except for the I/O bus, there are no time multiplexing of functions.

PIN CONFIGURATION



FEATURES

- **Fetch, Decode, and Execute a 16-bit Instruction in a minimum of 250-nanoseconds (one machine cycle)**
- **Bit-oriented Instruction set (addressable single-or-multiple bit subfields)**
- **Separate address, instruction, and I/O buses**
- **Source/destination architecture**
- **On-Chip oscillator and timing generation**
- **Eight 8-bit working registers**
- **TTL Inputs and outputs**
- **BiPolar Low-Power Schottky technology**
- **3-State I/O bus**
- **Single +5V supply**

ORDERING INFORMATION

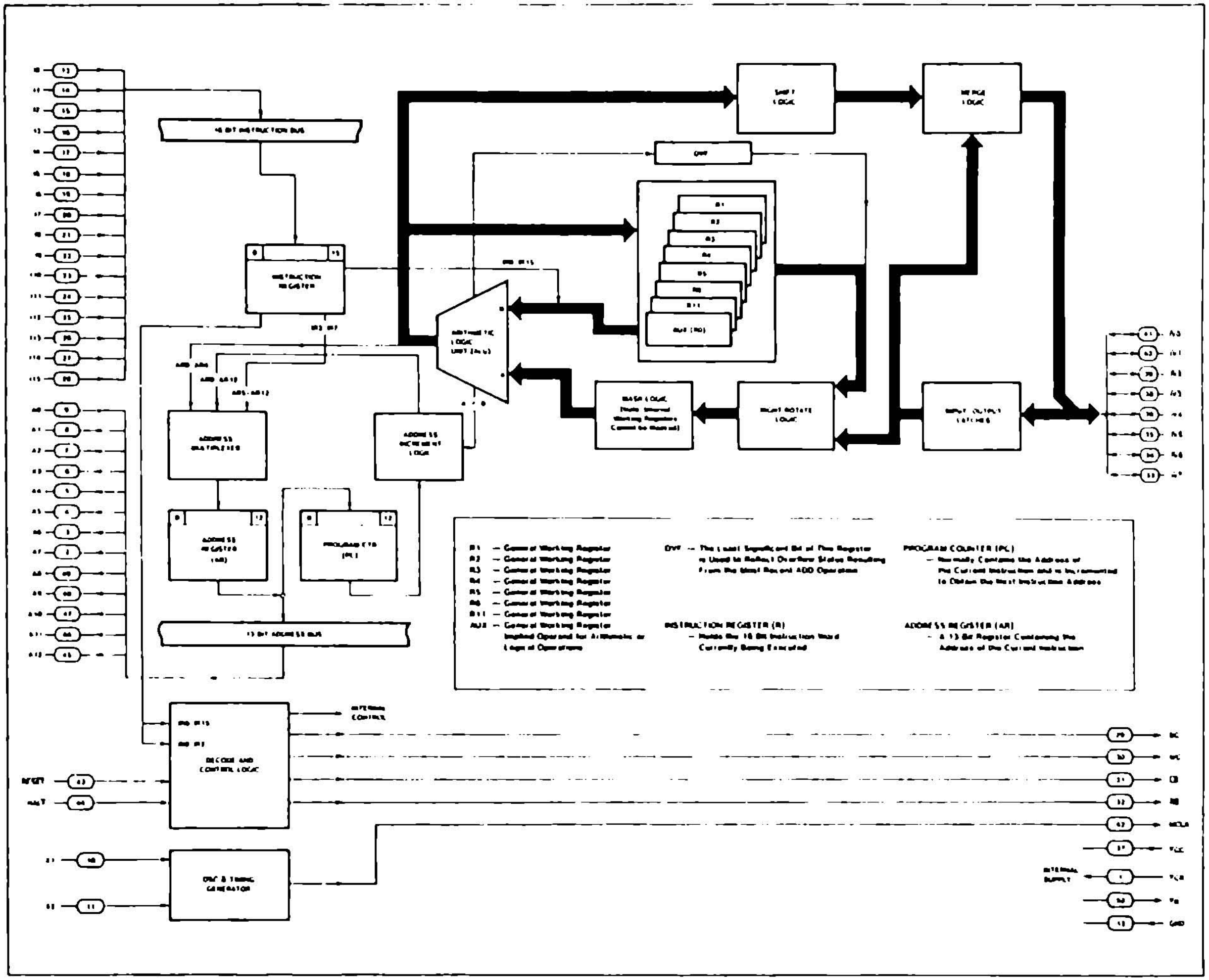
Commercial

Order number: N8X300I
Packaging information: Refer to Signetics price list
Supply voltage: 5V (± 5%)
Operating temperature range: 0°C to +70°C

Military

Order number: S8X300-1
Packaging information: Refer to Signetics price list
Supply voltage: 5V (± 5%)
Operating temperature range: −40°C to +100°C

Order number: S8X300-2
Packaging information: Refer to Signetics price list
Supply voltage: 5V (± 10%)
Operating temperature range: −20°C to +100°C


16 BIT INSTRUCTION BUS
INSTRUCTION REGISTER
SHIFT LOGIC
MERGE LOGIC
OVF
R1
R2
R3
R4
R5
R6
R11
AUX (R0)
ARITHMETIC LOGIC UNIT (ALU)
MASK LOGIC
RIGHT ROTATE LOGIC
INPUT OUTPUT LATCHES
ADDRESS MULTIPLEXER
ADDRESS INCREMENT LOGIC
ADDRESS REGISTER (AR)
PROGRAM CTR (PC)
13 BIT ADDRESS BUS
DECODE AND CONTROL LOGIC
EXTERNAL CONTROL
OSC & TIMING GENERATOR
RESET
HALT
X1
X2
SC
WC
LB
RB
MCLK
VCC
VCR
VR
GND
INTERNAL SUPPLY

R1 — General Working Register
R2 — General Working Register
R3 — General Working Register
R4 — General Working Register
R5 — General Working Register
R6 — General Working Register
R11 — General Working Register
AUX — General Working Register Implied Operand for Arithmetic or Logical Operations

OVF — The Least Significant Bit of This Register is Used to Reflect Overflow Status Resulting From the Most Recent ADD Operation

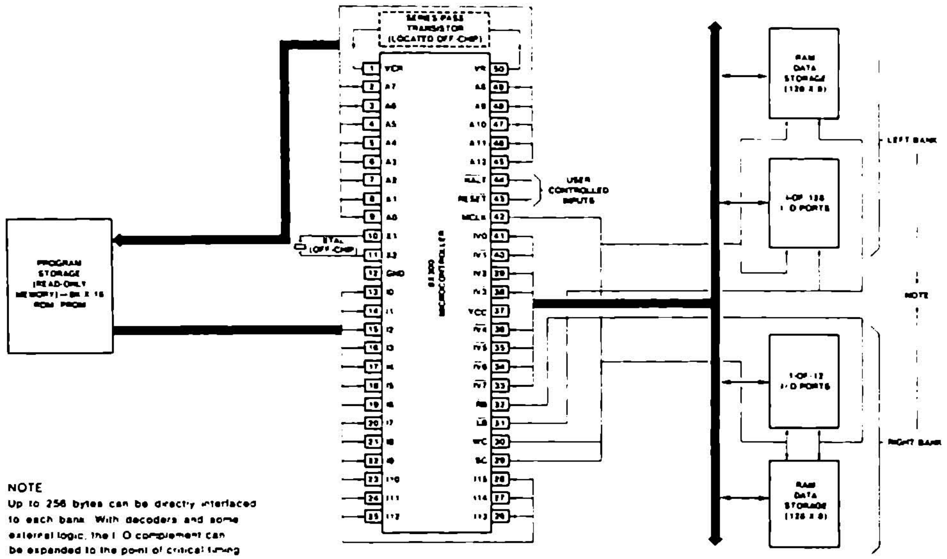
INSTRUCTION REGISTER (IR)
— Holds the 16 Bit Instruction Word Currently Being Executed

PROGRAM COUNTER (PC)
— Normally Contains the Address of the Current Instruction and is Incremented to Obtain the Next Instruction Address

ADDRESS REGISTER (AR)
— A 13 Bit Register Containing the Address of the Current Instruction

Figure 1. CPU Architecture and PIN Designations For 8X300 Microcontroller

VALVO


SERIES PASS TRANSISTOR (LOCATED OFF-CHIP)
PROGRAM STORAGE (READ-ONLY MEMORY)—8K X 16 ROM PROM
XTAL (OFF-CHIP)
8X300 MICROCONTROLLER
1 VCR
2 A7
3 A6
4 A5
5 A4
6 A3
7 A2
8 A1
9 A0
10 X1
11 X2
12 GND
13 I0
14 I1
15 I2
16 I3
17 I4
18 I5
19 I6
20 I7
21 I8
22 I9
23 I10
24 I11
25 I12
50 VR
49 A8
48 A9
47 A10
46 A11
45 A12
44 HALT
43 RESET
42 MCLK
41 IV0
40 IV1
39 IV2
38 IV3
37 VCC
36 IV4
35 IV5
34 IV6
33 IV7
32 RB
31 LB
30 WC
29 SC
28 I15
27 I14
26 I13
USER CONTROLLED INPUTS
RAM DATA STORAGE (128 X 8)
1-OF-128 I/O PORTS
LEFT BANK
NOTE
1-OF-12 I/O PORTS
RAM DATA STORAGE (128 X 8)
RIGHT BANK

NOTE
Up to 256 bytes can be directly interfaced to each bank. With decoders and some external logic, the I/O complement can be expanded to the point of critical timing

PIN NO.	IDENTIFIER	NAME AND FUNCTION	ACTIVE STATE
2-9/45-49	A0-A12	**Program Address Lines:** These outputs permit direct addressing of up to 8192 words of program storage. A high voltage level equals a binary "1". A12 is Least Significant Bit.	High
13-28	I0-I15	**Instruction Lines:** These input lines receive 16-bit instructions from program storage. A high voltage level equals a binary "1". I15 is Least Significant Bit	High
33-36 38-41	$\overline{IV0}$-$\overline{IV7}$	**Input/Output Bus:** These bidirectional three-state lines communicate with up to 512 I/O devices (256 per bank). A low voltage level equals a binary "1". $\overline{IV7}$ is Least Significant Bit	Low
10 & 11	X1 & X2	**Connections** for a capacitor, a series-resonant crystal, or an external clock source with complementary outputs. For precise frequency control, a crystal or external source is required	—
42	MCLK	**Master Clock:** This output is used for clocking I/O devices and/or synchronization of external logic	High
30	WC	**Write Command:** When signal is high (binary 1), data is being output on pins $\overline{IV0}$-$\overline{IV7}$ of I/O bus	High
29	SC	**Select Command:** When signal is high (binary 1), an address is being output on pins $\overline{IV0}$-$\overline{IV7}$ of I/O bus	High
31	$\overline{LB}$	When the $\overline{LB}$ signal is low (binary 0), any one of up-to-256 I/O devices (or memory locations) in the left bank can be accessed. When the address of a particular device (or memory location) matches the address on the IV bus, that particular device (or memory location) is enabled and selected for input/output operations. All addresses on the left bank that do not match are deselected	Low
32	$\overline{RB}$	When the $\overline{RB}$ signal is low (binary 0), any one of up-to-256 I/O devices (or memory locations) in the right bank can be accessed. When the address of a particular device (or memory location) matches the address on the I/O bus, that particular device (or memory location) is enabled and selected for input/output operations. All addresses on the right bank that do not match are deselected.	Low
43	$\overline{RESET}$	When reset input is low (binary 0), the microcontroller is initialized—sets Program Counter/Address to zero and inhibits MCLK output	Low
44	$\overline{HALT}$	When halt input is low (binary 0), internal operation of microcontroller stops at the start of next instruction. The stop function does not inhibit MCLK or affect any internal registers.	Low
50	VR	Internally-generated reference output voltage for external series-pass transistor	—
1	VCR	Regulated voltage input from series-pass transistor (2N5320 or equivalent)	—
12	GND	Circuit ground	—
37	V_{CC}	Input connection for +5V power	—

Figure 2. Typical 8X300 System with Pin Definitions

TYPICAL 8X300 SYSTEM HOOKUP

Although the system hookup shown in Figure 2 is of the simplest form, it provides a fundamental look at the 8X300 microcontroller and peripheral relationships. As indicated, program storage can be either ROM or PROM and, by using various addressing-methods/decoding-schemes, memory paging techniques can be easily implemented. Also, by proper bit assignment, some external interface logic and, under software control, the program memory can be used as a storage device for interrupt-service subroutines. The user interface ($\overline{IV0}$ through $\overline{IV7}$) is capable of addressing 256 Input/Output ports and, with the additional bank-select bit ($\overline{LB}$ and $\overline{RB}$), the number of addressable I/O ports is 512—the left bank and right bank each consisting of 256 ports. The I/O ports of each bank can be used in a variety of ways; one of these ways is shown in Figure 2. When $\overline{LB}$ is active low, the left bank can be enabled and, providing there is an address match, anyone of 128 I/O ports or anyone of 128 locations within the RAM memory can be accessed for input/output operations. When $\overline{RB}$ is active low, the same set of conditions are applicable to the right bank. With some sacrifice in speed, any given I/O port can be interfaced to a memory peripheral or other I/O device of the user.

PROGRAM STORAGE INTERFACE

As shown in Figure 2, program storage is connected to output address lines A0 through A12 (A12 = LSB) and input instruction lines I0 through I15. An address output on A0/A12 identifies one 16-bit instruction word in program storage. The instruction word is subsequently input on I0/I15 and defines the microcontroller operations which are to follow.

The Signetics 82S115 PROM or any TTL-compatible memory can be used for program storage. (Note. The worst-case access time depends upon the instruction cycle time, and also, the overall system configuration.)

I/O INTERFACE AND CONTROL

An 8-bit I/O data bus is used by the microcontroller to communicate with two fields of I/O devices. The complementary $\overline{LB}$ and $\overline{RB}$ signals identify which field of the I/O devices is enabled.

Both data and address information are output on the I/O bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and address information as follows:

SC	WC	FUNCTION
High	Low	I/O address is being output on the I/O (IV) bus
Low	High	I/O data is being output on the I/O (IV) bus
Low	Low	Input data expected from selected I/O device
High	High	Invalid (not generated by 8X300)

DATA PROCESSING

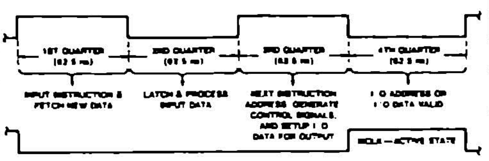
From a data processing point of view, the 8X300 microcontroller chip (Figure 1) contains eight 8-bit working registers (R1 through R6, R11, AUXiliary), an arithmetic logic unit (ALU), an overflow register (OVF), rotate/shift/mask/merge logic, and a bidirectional 8-bit I/O bus. Internal 8-bit data paths connect the registers and I/O bus to the ALU inputs, and the ALU output to the registers and I/O bus. Inputs to the ALU are preceded by the data-rotate and data-mask logic and the ALU output is followed by the shift and merge logic. Any one or all of the logic functions can operate on 8-bits of data in a single instruction cycle. Data from the source register can be right-rotated (end around) before processing by the ALU, external data (I/O bus) can also be masked to isolate a portion of the 8-bit field. Since the ALU always processes 8-bits of data, bit positions not specified by the mask operation are filled with zeroes.

When less than 8-bits of data are specified as output to the I/O bus from the ALU, the data field (shifted and masked, as required) is merged with prior contents of the I/O latches to form the output data. Bit positions of the I/O data not affected by the logic operations are not modified. Depending upon whether an I/O peripheral or an internal register is specified in the instruction as the source of data, the I/O latches contain, respectively, I/O-bus source data or destination data. For instance, when an internal register is specified as a source of data and an I/O peripheral as the destination, data from the peripheral is read into the I/O latches at the start of the instruction cycle, processed data is then merged with contents of the I/O latches to form the I/O output data at the end of the instruction cycle. When an I/O peripheral is specified as both data source and destination, data from the source is used both as the input to the I/O latches and as data to be processed; the processed data is then merged with data from the I/O latches to form the previously-described I/O bus output. If the data source and destination are on opposite banks of the 8X300 bus, the destination data is written with a full 8-bits, since the prior contents were not stored in the I/O latches.

INSTRUCTION CYCLE

Each microcontroller operation is executed in a single instruction cycle. The instruction cycle is divided into quarters with each quarter cycle being as short as 62.5-nanoseconds. Figure 3 shows the general functions that occur during each quarter cycle; specifics regarding minimum/maximum timing and other critical values are described under "Design Parameters" in this data sheet. During the first quarter cycle, a new instruction from program storage is input on signal lines I0 through I15; simultaneously, new data is fetched via the input/output bus ($\overline{IV0}$ through $\overline{IV7}$). At the end of the first quarter cycle, the new instruction is latched in the instruction register and the new I/O data is present at the input of the chip but is not, as yet, latched by the IV latches.

In the second quarter cycle, the I/O data stabilizes and preliminary processing is completed; at the end of this quarter, the IV latches are closed and final processing can be accomplished. During the third quarter cycle, the address for the next instruction is output to the I/O (IV) bus, control signals are generated, and I/O data is setup for the output phase. During the fourth quarter cycle, a master clock signal (MCLK) generated by the 8X300 is used to latch valid address or data into peripheral devices connected to the IV bus; MCLK is also used to synchronize any external logic with timing circuits of the 8X300. To summarize the action, the first half of the instruction cycle deals primarily with input functions and the second half is mostly concerned with output functions.


1ST QUARTER (62.5 ns)
2ND QUARTER (62.5 ns)
3RD QUARTER (62.5 ns)
4TH QUARTER (62.5 ns)
INPUT INSTRUCTION & FETCH NEW DATA
LATCH & PROCESS INPUT DATA
NEXT INSTRUCTION ADDRESS, GENERATE CONTROL SIGNALS, AND SETUP I/O DATA FOR OUTPUT
I/O ADDRESS OR I/O DATA VALID
MCLK—ACTIVE STATE

NOTES

1 New instruction must be accepted and latched at end of first quarter cycle
2 The I/O data latches are open for the first two quarter cycles, that is, for 125-nanoseconds
3 The address is output during third quarter cycle
4 I/O bus drivers are active (turned on) during third and fourth quarter cycles

Figure 3. Instruction Cycle and MCLK with: Crystal = 8MHz and Cycle Time = 250 ns

INSTRUCTION SET

General Format and Basic Operations

The 16-bit instruction word (I0 through 15) from program storage is input to the instruction register (Figure 1) and is subsequently decoded to implement the events to occur during the current instruction cycle. The instruction word is formatted as follows:

MSB ↓ LSB ↓

BIT POSITIONS →

0 1 2	3 4 5 6 7 8 9 10 11 12 13 14 15
OPCODE	OPERAND(S)

Rather than discrete instructions, the three operation code (OP CODE) bits specify eight instruction classes. Each instruction class is subject to a number of powerful variations, these variations are specified by the thirteen operand bits. General areas of control for the eight instruction classes are:

- Arithmetic and Logic Operations (ADD, AND, AND XOR)
- Movement of Data and Constants (MOVE and XMIT)
- Branch or Test (JMP, NZT, and XEC)

Basic operations for each of the eight instruction classes are as follows; a summary of the instruction set is provided in Table 1.

MOVE—data in source register or I/O-bus input is moved to destination register or I/O-bus output. Data can be shifted any number of places and/or masked to any length.

ADD—data in source register or I/O-but input is added to content of AUX (R0) register and the result is placed in the destination register or I/O-bus output. Data can be shifted and/or masked, as required.

Table 1. Summary of 8X300 Instruction Set

<table>
<tr><th rowspan="2">INSTRUC CLASS</th><th rowspan="2">OPCODE</th><th rowspan="2">FORMATS</th><th rowspan="2">DESCRIPTION</th><th rowspan="2">I/O CONT SIG</th><th colspan="2">STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE</th></tr>
<tr><th>INPUT PHASE (INSTRUCTION INPUT & DATA PROCESSING)</th><th>OUTPUT PHASE (ADDRESS & I/O BUS)</th></tr>
<tr><td>MOVE</td><td>0</td><td>F1 Register to Register<table><tr><td>0 1 2</td><td>3 4 5 6 7</td><td>8 9 10</td><td>11 12 13 14 15</td></tr><tr><td>OPCODE</td><td>S</td><td>R</td><td>D</td></tr></table>Invalid values of "S" 07_8, 17_8, 20_8-37_8
Invalid values of "D" 10_8, 20_8-37_8</td><td>(S) → D
Move content of internal register specified by S-field to internal register specified by D field. Prior to the "MOVE" operation, right rotate contents of internal source register by octal value (0 through 7) defined by the R field</td><td>SC =
WC =
$\overline{\text{LB}}$ =
$\overline{\text{LB}}$ =</td><td>L
L
X
X</td><td>H if "D" = 07_8, 17_8
L
H if "D" = 17_8
L if "D" = 07_8</td></tr>
<tr><td></td><td></td><td>F2 I/O Bus to Register<table><tr><td>0 1 2</td><td>3 4 5 6 7</td><td>8 9 10</td><td>11 12 13 14 15</td></tr><tr><td>OPCODE</td><td>S</td><td>L</td><td>D</td></tr></table>Valid values of "S" 20_8-37_8
Invalid values of "D" 10_8, 20_8-37_8</td><td>Move right-rotated I/O bus (source) data specified by the S field to internal register specified by the D-field. The L-field specifies the length of source data starting from the LSB-position and, if less than 8-bits, the remaining bits are filled with zeroes</td><td>SC =
WC =
$\overline{\text{LB}}$ =
$\overline{\text{LB}}$ =</td><td>L
L
L if "S" = 20_8-27_8
H if "S" = 30_8-37_8</td><td>H if "D" = 07_8, 17_8
L
H if "D" = 17_8
L if "D" = 07_8</td></tr>
<tr><td></td><td></td><td>F3: Register to I/O Bus<table><tr><td>0 1 2 3</td><td>4 5 6 7</td><td>8 9 10</td><td>11 12 13 14 15</td></tr><tr><td>OPCODE</td><td>S</td><td>L</td><td>D</td></tr></table>Invalid values of "S" 07_8, 17_8, 20_8, 37_8
Valid values of "D" 20_8-37_8</td><td>Move contents of internal register specified by the S-field to the I/O latches. Before outputting on I/O bus, data is shifted as specified by the least significant octal digit of the D-field and the bits specified by the L-field are merged with the latched I/O data</td><td>SC =
WC =
$\overline{\text{LB}}$ =
$\overline{\text{LB}}$ =</td><td>L
L
L if "D" = 20_8-27_8
H if "D" = 30_8-37_8</td><td>L
H
L if "D" = 20_8-27_8
H if "D" = 30_8-37_8</td></tr>
<tr><td></td><td></td><td>F3: I/O Bus to I/O Bus<table><tr><td>0 1 2</td><td>3 4 5 6 7</td><td>8 9 10</td><td>11 12 13 14 15</td></tr><tr><td>OPCODE</td><td>S</td><td>L</td><td>D</td></tr></table>Valid values of "S" 20_8-37_8
Valid values of "D" 20_8-37_8</td><td>Move right rotated I/O-bus (source) data specified by the S-field to the I/O latches. Before outputting on I/O bus, shift data as specified by the D-field, then merge source and latched I/O data as specified by the L (length) field</td><td>SC =
WC =
$\overline{\text{LB}}$ =
$\overline{\text{LB}}$ =</td><td>L
L
L if "D" = 20_8-27_8
H if "D" = 30_8-37_8</td><td>L
H
L if "D" = 20_8-27_8
H if "D" = 30_8-37_8</td></tr>
</table>

Table 1. Summary of 8X300 Instruction Set (Continued)

INSTRUC CLASS	OPCODE	FORMATS	DESCRIPTION	I/O CONT SIG	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE	
					INPUT PHASE (INSTRUCTION INPUT & DATA PROCESSING)	OUTPUT PHASE (ADDRESS & I/O BUS)
ADD	1	Same as MOVE instruction class	(S) plus (AUX) → D Same as MOVE instruction class except that contents of AUX (R0) register are ADDed to the source data. If there is a "carry" from MSB, then OVF (overflow) = 1, otherwise OVF = 0	Same as MOVE instruction class		
AND	2	Same as MOVE instruction class	(S) $\wedge$ (AUX) → D Same as MOVE instruction class except that contents of AUX (R0) register are ANDed with source data	Same as MOVE instruction class		
XOR	3	Same as MOVE instruction class	(S) $\oplus$ (AUX) → D Same as MOVE instruction class except that contents of AUX (R0) register are exclusively ORed with source data	Same as MOVE instruction class		
XEC	4	**F3: Register Immediate** 0 1 2 | 3 4 5 6 7 | 8 9 10 11 12 13 14 15 OPCODE | S | J Invalid values of "S" 07_8, 17_8, 20_8-37_8 Valid values of "J" 000_8-377_8	**Execute instruction at current page address offset by J (literal) + (S). Return to normal instruction flow unless a branch is encountered.** Execute instruction at an address determined by replacing the low order 8 bits of the Program Counter with the following derived sum • Value of literal (J field) plus • Contents of internal register specified by S field The PC is not incremented and the overflow status (OVF) is not changed	SC = WC = $\overline{LB}$ =	L L X	L L X
		F4: I/O Bus Immediate 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | J Valid values of "S" 20_8-37_8 Valid values of "J" 00_8-37_8	Execute instruction at an address determined by replacing the low order 5 bits of Program Counter with the following derived sum • 5-bit value of literal (J field) plus • Value of rotated source data specified by S field (The L field specifies the length of source data starting from the LSB position and if less than 8 bits, the remaining bits are filled with zeros. The Program Counter is not incremented and the overflow status (OVF) is not changed.)	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "S" = 20_8-27_8 H if "S" = 30_8-37_8	L L X X
NZT	5	**F3: Register Immediate** 0 1 2 | 3 4 5 6 7 | 8 9 10 11 12 13 14 15 OPCODE | S | J Invalid values of "S" 07_8, 17_8, 20_8-37_8 Valid values of "J" 000_8-377_8	**If data specified by the S-field is not equal to zero, jump to current page address offset by value of J-field, otherwise, increment the Program Counter.** If contents of internal register specified by S field is non-zero, transfer to address determined by replacing the low order 8 bits of Program Counter with "J", otherwise increment PC	SC = WC = $\overline{LB}$ =	L L X	L L X
		F4: I/O Bus Immediate 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | J Valid values of "S" 20_8-37_8 Valid values of "J" 00_8-37_8	If right rotated I/O bus data is non-zero, transfer to address determined by replacing low order 5 bits of Program Counter with "J", otherwise, increment PC. (The L field specifies the length of source I/O data starting from the LSB position and if less than 8 bits, the remaining bits are filled with zeroes.)	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "S" = 20_8-27_8 H if "S" = 30_8-37_8	L L X X
XMIT	6	**F3: Register Immediate** 0 1 2 | 3 4 5 6 7 | 8 9 10 11 12 13 14 15 OPCODE | D | J Invalid values of "D" 10_8, 20_8-37_8 Valid values of "J" 000_8-377_8	Transmit J → D Transmit and store 8-bit binary pattern in J field to internal register specified by D field	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L X X	H if D = 07_8 or 17_8 L H if D = 17_8 L if D = 07_8
		F4: I/O Bus Immediate 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | D | L | J Valid values of "D" 20_8-37_8 Valid values of "J" 00_8-37_8	Transmit binary pattern in J field to I/O bus. Before putting data on I/O bus, shift literal value "J" as specified by the D field and merge bits specified by the L field with existing I/O bus data. If the L field specifies more than 5 bits starting from the LSB position, all remaining bits are set to zero	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if D = 20_8-27_8 H if D = 30_8-37_8	L H L if D = 20_8-27_8 H is D = 30_8-37_8
JMP	7	**F5: Address Immediate** 0 1 2 | 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | A Valid values of A 00000_8-17777_8	Jump to address in program storage specified by A field, thus address is loaded into the Address Register and the Program Counter	SC = WC = $\overline{LB}$ =	L L X	L L X

NOTES

* $\overline{RB}$ is complement of $\overline{LB}$, X = Undefined

AND—data in source register or I/O-bus input is ANDed with content of AUX (R0) register and the result is placed in the destination register or I/O-bus output. Data can be shifted and/or masked, as required.

XOR—data in source register or I/O-bus input is exclusively ORed with contents of AUX (R0) register and the result is placed in the destination register or I/O-bus output. Data can be shifted and/or masked, as required.

XMIT—immediate data field of instruction word replaces data in destination register or I/O-bus output.

XEC—executes instruction at the program address which is formed by replacing the least significant bits of the last address with the sum of:

- Literal (J) field value of instruction plus,
- Value of data in source register or I/O-bus input.

NZT—least significant bits of program address are replaced by literal (J) field of instruction if the source register or I/O-bus is not equal to zero.

JMP—program address is replaced by address field of the instruction word.

Instruction Fields

As shown in Table 1, each instruction contains an operations code (OPCODE) field and from one-to-three operand fields. The operand fields are: Source (S), Destination (D), Rotate/Length (R/L), Literal (J), and Address (A). The OPCODE and operand fields are briefly described in the following paragraphs.

Operations Code Field: The three-bit OPCODE field specifies one of eight classes of 8X300 instructions; octal designations for this field and operands for each instruction class are shown in Table 1.

Source (S) and Destination (D) Fields: The five-bit (S) and (D) fields specify the source and destination of data for the operation defined by the OPCODE field. The AUXiliary (R0) register is an implied second operand for the ADD, AND, and XOR instructions, each of which require two source fields. That is, instructions of the form:

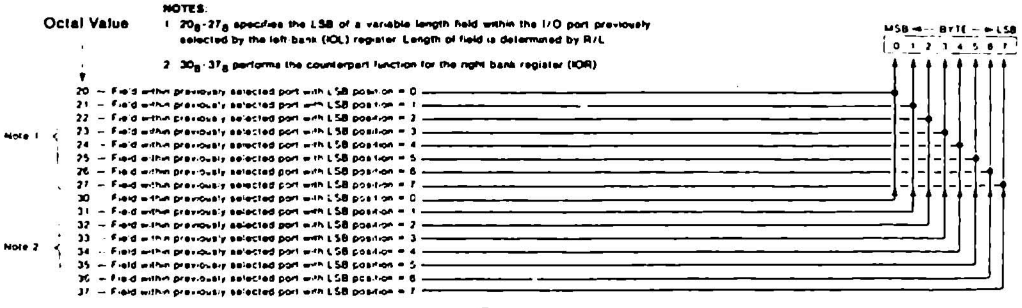
ADD X, Y

imply a third operand, say Z, located in the AUX (R0) register. Thus, the operation for the preceding expression is actually (X + Z), with the result stored in Y. The (S) and/or (D) fields can specify an internal 8X300 register or any one-to-eight bit I/O field; octal values for these registers and Source/Destination field assignments are provided in Table 2.

Table 2. Octal Addresses of 8X300 Registers and Address/Bit Assignments of Source/Destination Fields

Octal Value	8X300 Register	Octal Value	8X300 Register
00	Auxiliary (R0)	10	OVF (Overflow Register)—used only as a source
01	R1	11	R11
02	R2	12	Unassigned
03	R3	13	Unassigned
04	R4	14	Unassigned
05	R5	15	Unassigned
06	R6	16	Unassigned
07	*IOL Register—Left Bank I/O Address Register; Used only as destination	17	*IOR Register—Right Bank I/O Address Register; Used only as destination

NOTE
*If IOL or IOR is specified as a source of data, the source data is all zeroes



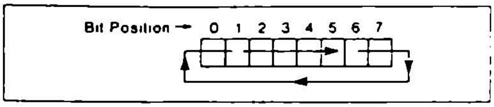
NOTES:
1 20_8-27_8 specifies the LSB of a variable length field within the I/O port previously selected by the left-bank (IOL) register. Length of field is determined by R/L
2 30_8-37_8 performs the counterpart function for the right bank register (IOR)

Octal Value
MSB ← BYTE → LSB
0 1 2 3 4 5 6 7
Note 1
20 — Field within previously selected port with LSB position = 0
21 — Field within previously selected port with LSB position = 1
22 — Field within previously selected port with LSB position = 2
23 — Field within previously selected port with LSB position = 3
24 — Field within previously selected port with LSB position = 4
25 — Field within previously selected port with LSB position = 5
26 — Field within previously selected port with LSB position = 6
27 — Field within previously selected port with LSB position = 7
Note 2
30 — Field within previously selected port with LSB position = 0
31 — Field within previously selected port with LSB position = 1
32 — Field within previously selected port with LSB position = 2
33 — Field within previously selected port with LSB position = 3
34 — Field within previously selected port with LSB position = 4
35 — Field within previously selected port with LSB position = 5
36 — Field within previously selected port with LSB position = 6
37 — Field within previously selected port with LSB position = 7

Rotate (R) and Length (L) Field: The three-bit R/L field performs one of two functions, specifying either the field length (L) or a right-rotate (R). For a given instruction, the specified function depends upon the contents of the source (S) and destination (D) fields.

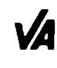
When an internal register is specified by both the source and destination fields, the (R) field is invoked and it specifies a right-rotate of the data specified in the (S) field—see accompanying diagram. The source-register data (up to eight-bits) is right-rotated within one instruction cycle. (The right-rotate function is implemented on the bus and not in the source register.)

RIGHT-ROTATE FUNCTION



When either or both of the source and destination fields specify a variable-length I/O data field, the (L) field specifies the length of the I/O data field—see accompanying diagram. If the source field specifies an I/O address (20_8-37_8) and the destination field specifies an internal register (00_8-06_8, 07_8, 11_8, or 17_8), the L-field specifies the length of source data, the source data is formed by right rotating the I/O bus data according to the source address (Table 2) and then masking result as specified by L-field. If length is less than eight-bits, all remaining bits are set to zero prior to processing data in the ALU. If the source field specifies an internal register (00_8,-06_8, 10_8,or 11_8) and the destination field specifies I/O bus data (20_8-37_8), the L field specifies the length of the destination data. To form the destination data, the ALU output is left-shifted according to the destination address (Table 2) and then masked to the required length—see DATA LENGTH SPECIFICATION. The destination data is merged with data in the I/O latches to finalize the I/O bus data. Hence, a one-to-eight bit destination data field can be inserted into the existing eight-bit I/O port without modifying surrounding bits. If both the source and destination fields specify I/O bus data (20_8-37_8), the L-field specifies the length of both the source and destination data.

DATA LENGTH SPECIFICATION



To form the source data, the I/O bus data is right-rotated according to the source address (Table 2) and then masked to the required length—see preceding DATA LENGTH SPECIFICATION. If length is less than eight-bits, all remaining bits are set to zero before processing in the ALU. To form the destination data, the ALU output is left-shifted according to the destination address (Table 2) and masked to the required length specification. The destination data is then merged into the I/O bus data that was used to obtain the source; thus, if the source and destination addresses are on the same bank, the I/O bus data written to the destination register appears unmodified, except for bits changed during the shift-and-mask operations. If the source and destination addresses refer to different banks, the destination register is changed to contain the contents of the source register in those bit positions not affected by the destination data.

J-Field: The 5-bit or 8-bit (J) field is used to load a literal value (contained in the instruction) into a register, into a variable I/O data field, or to modify the low-order bits of the Program Counter. The bit-length of the (J) field is implied by the (S) field in the XEC, NZT, and XMIT instructions, based on the following considerations.

- When the source (S) field specifies an internal register, the literal value of the J-field is an 8-bit binary number.
- When the source (S) field specifies a variable I/O data field, the literal value of the J-field is a 5-bit binary number.

A-Field: The 13-bit (A) field is an address field which allows the 8X300 to directly address up 8192 locations in Program Storage memory.

INSTRUCTION SEQUENCE CONTROL

Formation of Instruction Address

The Address Register and Program Counter are used to generate addresses for accessing an instruction from program storage. The instruction address is formed in any one of four ways:

- For all except the JMP, XEC, and a "satisfied" NZT instruction, the Program Counter is incremented by one and placed in the Address Register.
- For the JMP instruction, the 13-bit A-field contained in the JMP instruction word replaces the contents of both the Address Register and Program Counter.
- For the XEC instruction, the Address Register is loaded with the high-order bits of the Program Counter modified as follows:

  XEC using I/O Bus Data: low order 5-bits of ALU output replaces counterpart bits in Address Register.

  XEC using Data from Internal Register: low order 8-bits of ALU output replaces counterpart bits in Address Register.

  The Program Counter is not modified for either of the above conditions.
- For a "satisfied" NZT instruction, the low order 5-bits (NZT source is I/O Bus Data) or low order 8/bits (NZT source is an Internal Register) of both the Address Register and Program Counter are loaded with the literal value specified by J-field of the instruction word.

Data Addressing

The source and/or destination addresses of the data to be operated upon are specified as part of the instruction word. As shown in Table 3, source/destination addresses are specified using a five-bit address (00_8 through 37_8). When the most significant octal digit is a 0 or 1, the source and/or destination address is an internal register; if the most significant digit is a 2 or 3, an I/O bus address is indicated—2 specifying a left-bank (LB) address and 3 specifying a right-bank (RB) address. The least significant octal digit (0 through 7) indicates either a specific internal register address or positioning information for the least significant bit when specifying I/O bus data. Referring to Table 1, the AUXiliary register (00) is the implied source of the second argument for the ADD, AND, and XOR operations. IOL (destination address 07_8) and IVR (destination address 17_8) provide a means of routing address information to I/O registers. With IOL or IOR specified as the destination address, the data is placed on the I/O bus during the output phase of the instruction cycle. Simultaneously, a select command (SC) is generated to inform all I/O devices that information on the I/O bus is to be considered as an I/O address. Since IOL and IOR are not harware registers, they should never be specified as a source address.

Control outputs $\overline{LB}$ and $\overline{RB}$ are used to partition I/O bus devices into two fields of 256 addresses. With LB in the active-low state and a source address of 20_8-27_8, the left bank of I/O devices are enabled during the input phase of the instruction cycle. With $\overline{RB}$ in the active-low state and a source address of 30_8-37_8, the right bank of devices are enabled. During the output phase, $\overline{RB}$ is low if the destination address is IOR (17_8) or 30_8-37_8; $\overline{LB}$ is low if the destination address is IOL (07_8) or 20_8-27_8. Each address field ($\overline{LB}$ and $\overline{RB}$) can have a different I/O device selected; thus two devices can be directly accessed within one instruction cycle.

Table 3. Source/Destination Addresses

Source and/or Destination Field (Octal)	Source/Destination
00	AUXiliary register (R0)
01-06	Working registers R1-R6, respectively
07	IOL Left-bank enable (Destination only)
10	Overflow status—OVF (Source only)
11	Working register R11
17	IOR Right-bank enable (Destination only)
2N (N = 0, 1, 2, 3 4, 5, 6, or 7)	If a source, I/O data is right-rotated (7 − N) bits and then masked as specified by the L-field $\overline{LB}$ = low and $\overline{RB}$ = high generated during input phase. If a destination, I/O data is left-shift (7 − N) bits and merged (specified by L-field) with data contained in the I/O latches $\overline{LB}$ = low and $\overline{RB}$ = high generated during output phase
3N (N = 0, 1, 2, 3, 4, 5, 6, or 7)	If a source, I/O data is right-rotated (7 − N) bits and then masked as specified by the L-field $\overline{LB}$ = high and $\overline{RB}$ = low generated during input phase If a destination, I/O data is left-shifted (7 − N) bits and merged (specified by L-field) with data contained in the I/O latches $\overline{LB}$ = high and $\overline{RB}$ = low generated during output phase

---

DESIGN PARAMETERS

Hardware design of an 8X300-based system largely consists of the following operations:

- **Selecting and Interfacing a Program Storage device—ROM, PROM, etc. (Pins 2 through 9 and 45 through 49 for 13-bit address interface; Pins 13 through 28 for 16-bit instruction interface.)**
- **Selecting and Interfacing Input/Output devices—RAM, Multiplexers, I/O Ports, and other eight-bit addressable I/O devices. (Pins 33 through 36 and pins 38 through 41 for eight-bit I/O interface.)**
- **Choosing and Implementing System Clock—Capacitor-Controlled, Crystal-Controlled, or Externally-Driven. (Pins 10 and 11 for System Clock Interface.)**
- **Selection of 5-volt power supply and off-chip series-pass transistor.**
- **External logic, as required, to meet the control requirements of a particular application.**

All information required for easy implementation of these design requirements is provided under the following captions.

- **DC Characteristics**
- **AC Characteristics**
- **Timing Considerations**
- **Clock Considerations**
- **HALT/RESET Logic**
- **Voltage Regulator**

DC CHARACTERISTICS (Commercial Part) 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT	COMMENTS
			Min	Typ	Max		
V_{CC}	Supply voltage		4.75	5.0	5.25	V	5V ± 5%, pin 37 only
V_{IH}	High level input voltage		0.6 2.0		2.0	V	X1 and X2 All other pins
V_{IL}	Low level input voltage				0.4 0.8	V	X1 and X2 All other pins
V_{OH}	High level output voltage	V_{CC} = min, I_{OH} = −3mA	2.4	3.0		V	
V_{OL}	Low level output voltage	V_{CC} = min, I_{OL} = 6mA V_{CC} = min, I_{OL} = 16mA		0.39 0.39	0.55 0.55	V	A0 through A12 All other outputs
V_{CR}	Regulator voltage	V_{CC} = 5V		3.1		V	From series-pass transistor
V_{IC}	Input clamp voltage	V_{CC} = min; I_{IN} = −10mA			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes.
I_{IH}	High-level input current	V_{CC} = max; V_{IH} = 0.6V V_{IH} = 4.5V	1.0	1.7 1	3.0 50	mA µA	X1 and X2 All other pins
I_{IL}	Low-level input current	V_{CC} = max, V_{IL} = 0.4V		 −0.13 −0.67 −0.23	−3 −0.2 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$-$\overline{IV7}$ I0-I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS}	Short circuit output current	V_{CC} = max, V_{CR} = V_{CRH} (Note. At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC}	Supply current	V_{CC} = max, V_{CR} = V_{CRH}			150	mA	
I_{REG}	Regulator control	V_{CC} = 5.0V	−14		−21	mA	
I_{CR}	Regulator current	V_{CC} = max			250	mA	

NOTES

1 Operating temperature ranges are guaranteed after thermal equilibrium has been reached

2 All voltages measured with respect to ground terminal

AC CHARACTERISTICS (Commercial Part)

CONDITIONS: V_{CC} = 5V (±5%), V_{IN} = 0V or 3V, 0°C ≤ T_A ≤ 70°C
LOADING: (See test circuits)

	PARAMETER (NOTE 1)	LIMITS (INSTRUCTION CYCLE TIME = 250 ns)			LIMITS (INSTRUCTION CYCLE TIME > 250 ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{PC}	Processor cycle time	250			250			ns	
T_{CP}	X1 clock period	125			125			ns	
T_{CH}	X1 clock high time	62			62			ns	
T_{CL}	X1 clock low time	62			62			ns	
T_{MCH}	MCLK high delay	31	42	52	31	42	52	ns	
T_{MCL}	MCLK low delay	31	42	52	31	42	52	ns	
T_W	MCLK pulse width	55	62	69	$T_{4Q}-7$	T_{4Q}		ns	Note 2
T_{AS}	X1 falling edge to address stable	50	63	80	50	63	80	ns	Note 7

AC CHARACTERISTICS (Commercial Part) (Continued)

CONDITIONS: V_{CC} = 5V (±5%), V_{IN} = 0V or 3V, 0°C ≤ T_A ≤ 70°C
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 250 ns)			LIMITS (INSTRUCTION CYCLE TIME > 250 ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{MAS}	MCLK falling edge to address stable	130	143	160	$T_{1Q}+T_{2Q}+5$	$T_{1Q}+T_{2Q}+18$	$T_{1Q}+T_{2Q}+35$	ns	Notes 2, 3, & 7
T_{IA}	Instruction to address			170			$T_{2Q}+108$	ns	Notes 2, 3, & 8
T_{IVA}	Input data to address			105			105	ns	Notes 3 & 9
T_{IS}	Instruction set-up time (X1 rising edge)	−7			−7			ns	Note 10
T_{MIS}	MCLK falling edge to instruction stable			20			$T_{1Q}-42$	ns	Notes 2, 4, & 10
T_{IH}	Instruction hold time (X1 rising edge)	45			45			ns	Note 11
T_{MIH}	Instruction hold time) (MCLK falling edge)	60			$T_{1Q}-2$			ns	Notes 2 & 11
T_{WH}	X1 falling edge to SC/WC rising edge	40	49	58	40	49	58	ns	
T_{MWH}	MCLK falling edge to SC/WC rising edge	125	130	135	$T_{1Q}+T_{2Q}$	$T_{1Q}+T_{2Q}+5$	$T_{1Q}+T_{2Q}+10$	ns ns	Note 2
T_{WL}	X1 falling edge to SC/WC falling edge	40	49	58	40	49	58	ns	
T_{MWL}	MCLK falling edge to SC/WC falling edge	5	7	15	5	7	15	ns	
T_{IBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	48	60	70	48	60	70	ns	
T_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	7	17	25	7	17	25	ns	
T_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)		27	35		27	35	ns	
T_{OBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	48	60	70	48	60	70	ns	
T_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	132	137	147	$T_{1Q}+T_{2Q}+7$	$T_{1Q}+T_{2Q}+12$	$T_{1Q}+T_{2Q}+22$	ns	Note 2
T_{IDS}	Input data set-up time (X1 falling edge)	25	16		25	16		ns	
T_{MIDS}	MCLK falling edge to input data stable		65	55		$T_{1Q}+T_{2Q}-60$	$T_{1Q}+T_{2Q}-70$	ns	Notes 2 & 5
T_{IDH}	Input data hold time (X1 falling edge)	40	30		40	30		ns	
T_{MDIH}	Input data hold time (MCLK falling edge)	125	112		$T_{1Q}+T_{2Q}$	$T_{1Q}+T_{2Q}-13$		ns	Note 2
T_{ODH}	Output data hold time (X1 falling edge)	55	65	75	55	65	75	ns	
T_{MODH}	Output data hold time (MCLK falling edge)	11	20	25	11	20	25	ns	
T_{ODS}	Output data stable (X1 falling edge)	74	84	94	74	84	94	ns	Notes 12, 14, & 15
T_{MODS}	Output data stable (MCLK falling edge)	150	160	170	$T_{1Q}+T_{2Q}+25$	$T_{1Q}+T_{2Q}+35$	$T_{1Q}+T_{2Q}+45$	ns	Notes 2, 12, 14, & 15

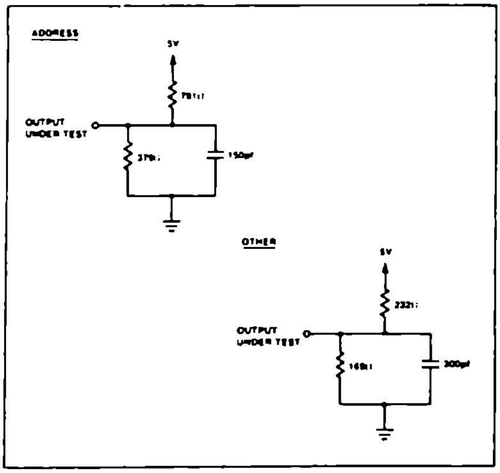
AC CHARACTERISTICS (Commercial Part) (Continued)

CONDITIONS: V_{CC} = 5V (± 5%), V_{IN} = 0V or 3V, 0°C ≤ T_A ≤ 70°C
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 250 ns)			LIMITS (INSTRUCTION CYCLE TIME > 250 ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{DD}	Input data to output data	104	120	136	104	120	136	ns	Notes 13 & 15
T_{HS}	$\overline{\text{HALT}}$ set-up time (X1 rising edge)	0			0			ns	
T_{MHS}	MCLK falling edge to $\overline{\text{HALT}}$ falling edge			18			$T_{1Q}-44$	ns	Notes 2 & 6
T_{HH}	$\overline{\text{HALT}}$ hold time (X1 rising edge)	32			32			ns	
T_{MHH}	$\overline{\text{HALT}}$ hold time (MCLK falling edge)	50			$T_{1Q}-12$			ns	Note 2
T_{ACC}	Program storage access time			80					
T_{IO}	I/O port output enable time ($\overline{\text{LB}}$/$\overline{\text{RB}}$ to valid IV data input)			30					

NOTES
1. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts; all timing parameters are measured at this voltage level.
2. Respectively T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
3. Capacitive loading for the address bus is 150 picofarads.
4. Same as TIS but referenced to falling edge of MCLK.
5. Same as TIDS but referenced to falling edge of MCLK.
6. Same as THS but referenced to falling edge of MCLK.
7. TAS is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set-up time. The TAS parameter then represents the earliest time that the address bus is valid.
8. TIA is obtained by forcing a valid instruction input to occur earlier than the minimum set-up time.
9. TIVA is obtained by forcing a valid I/O bus input to just meet the minimum set-up time.
10. TMIS represents the set-up time required by internal latches of the 8X300. In system applications, the instruction input may have to be valid before the worst-case set-up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set-up time (TIDS and TMIDS).
11. TIH represents the hold time required by internal latches of the 8X300. To generate proper LB/RB signals, the instruction must be held valid until the address bus changes.
12. TOOS is obtained by forcing a valid I/O bus input to occur earlier than the I/O bus input set-up time (TIDS). This timing parameter represents the earliest time that the I/O output data can be valid.
13. TOO is obtained by forcing a valid I/O bus input to just meet the minimum I/O bus input set-up time. Thus timing parameter represents the latest time that the I/O output data can be valid.
14. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the 8X300 will turn on.
15. For TIDS ≥ 35 ns, TOOS or TMOOS should be used to determine when the output data is stable.

TEST CIRCUITS



DC CHARACTERISTICS (Military Part)

S8X300-1 $-40°C \leq T_C \leq 100°C$ $V_{CC} = 5V \pm 5\%$
S8X300-2 $-20°C \leq T_C \leq 100°C$ $V_{CC} = 5V \pm 10\%$

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High level input voltage					
	X1, X2		0.6			V
	All others		2.0			V
V_{IL}	Low level input voltage					
	X1, X2				0.4	V
	All others				0.8	V
V_{IC}	Input clamp voltage (Notes 1 & 5)	V_{CC} = min I_I = −10mA			−1.5	V
I_{IH}	High level input current					
	X1, X2	V_{CC} = max V_{IH} = 0.6V			3.0	mA
	All others	V_{CC} = max V_{IH} = 4.5V			0.05	
I_{IL}	Low level input current					
	X1, X2	V_{CC} = max V_{IL} = 0.4V			−3.0	mA
	$\overline{IV0}$-$\overline{IV7}$	V_{CC} = max V_{IL} = 0.4V			−0.3	mA
	I0-I15	V_{CC} = max V_{IL} = 0.4V			−1.6	mA
	$\overline{HALT}$, $\overline{RESET}$	V_{CC} = max V_{IL} = 0.4V			−0.4	mA
V_{OL}	Low level output voltage					
	A0-A12	V_{CC} = min I_L = 4.25mA			0.55	V
	All others	V_{CC} = min I_{OL} = 16mA			0.55	V
V_{OH}	High level output voltage	V_{CC} = min I_{OH} = −3mA	2.4			V
I_{OS}	Short circuit output current (Note 2)	V_{CC} = max	−30		−140	mA
I_{CC}	Supply current (Note 4)	V_{CC} = max			160	mA
I_{REG}	Regulator control	V_{CC} = 5.0V	−14		−21	mA
I_{CR}	Regulator current	V_{CC} = max			285	mA
I_{CR}	Regulator current	$T_C \geq 25°C$ V_{CC} = max			330	mA
V_{CR}	Regulator voltage	$T_C < 25°C$ (Note 3)		3.1		V

NOTES

1. Crystal inputs X1 and X2 do not have clamp diodes
2. Only one output may be grounded at a time
3. From series-passed transistor under the following conditions
   VCC = Max HALT = RESET = ADDRESS = IVX = 0.0V, all other pins open
4. Pin 37 only
5. Test each input one at a time
6. All voltages are with respect to ground terminal
7. The operating temperature ranges are guaranteed after thermal equilibrium has been reached
8. Storage temperature −65°C to +150°C

AC CHARACTERISTICS (Military Part)

CONDITIONS: S8X300-1 — V_{CC} = 5V (± 5%) -40°C ≤ T_C ≤ 100°C
S8X300-2 — V_{CC} = 5V (± 10%) -20°C ≤ T_C ≤ 100°C

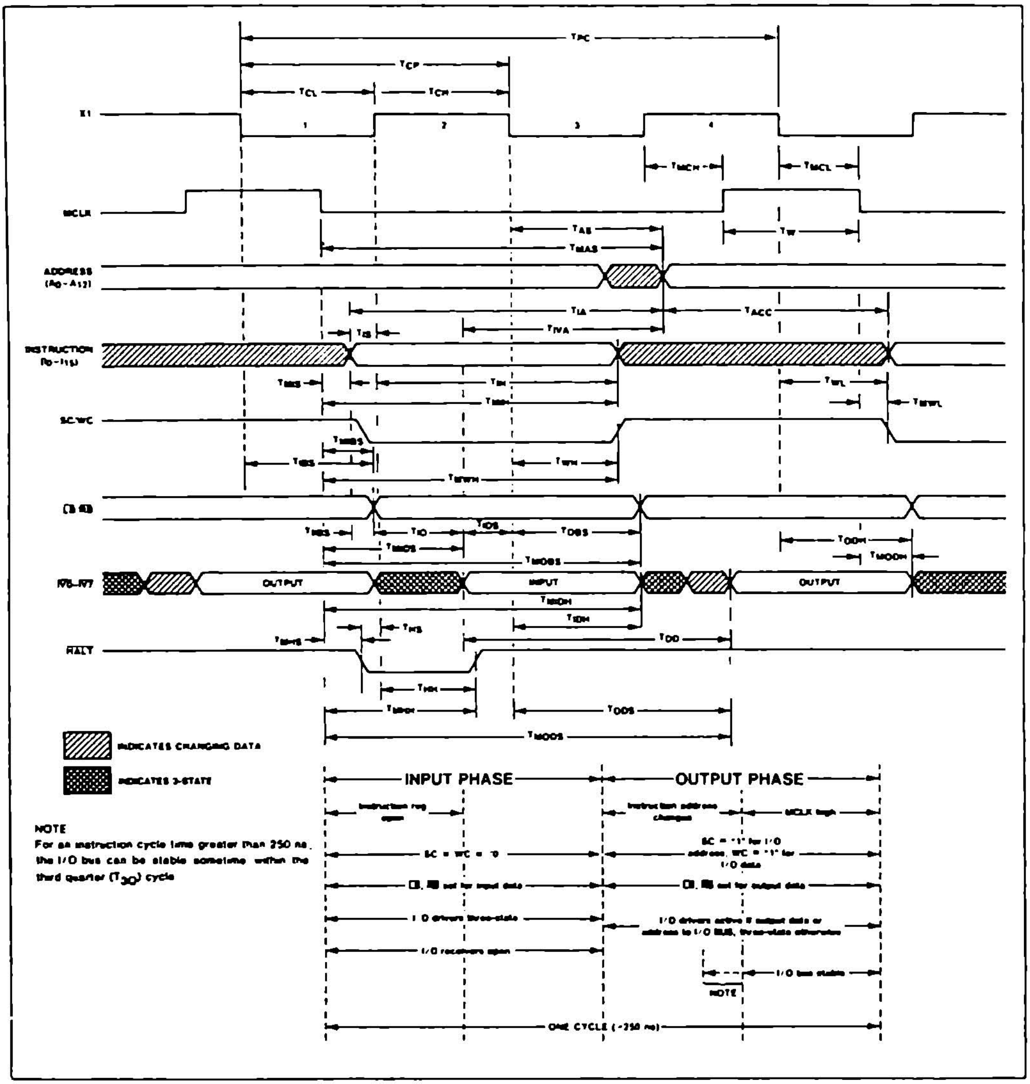
PARAMETER		TEST CONDITIONS (NOTES 1 & 2)	LIMITS			UNIT
			Min	Typ	Max	
Clock:						
T_{PC}	Processor cycle time		300			ns
T_{CP}	X1 clock period		150			ns
T_{CH}	X1 clock high time		62			ns
T_{CL}	X1 clock low time		62			ns
Controls:						
T_{HS}	$\overline{\text{HALT}}$ set-up time (X1 rising edge)		0			ns
T_{HH}	$\overline{\text{HALT}}$ hold time (X1 rising edge)		50			ns
Instructions:						
T_{AS}	X1 falling edge to address stable	CL = 100pF	35		92	ns
T_{IS}	Instruction set-up time (X1 rising edge)		0			ns
T_{IH}	Instruction hold time (X1 rising edge)		50			ns
T_{MCH}	MCLK high delay	X1 = 2.0V	20		55	ns
T_{MCL}	MCLK low delay	X1 = 2.0V	20		55	ns
T_{WH}	X1 falling edge to SC/WC rising edge				80	ns
T_{WL}	X1 falling edge to SC/WC falling edge				80	ns
T_{IIBS}	Instruction to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase)				52	ns
T_{IBS}	X1 falling edge to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase)		24			ns
T_{OBS}	X1 falling edge to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (output phase)				90	ns
T_{IDS}	Input data set-up time (X1 falling edge)		36			ns
T_{IDH}	Input data hold time (X1 falling edge)		50			ns
T_{ODS}	Output data stable (X1 falling edge)				125	ns
T_{ODH}	Output data hold time (X1 falling edge)		35		85	ns
T_{ACC}	Instruction access time	Provided by worst case timing	80			ns
T_{IO}	Data I/O access time	Provided by worst case timing	40			ns

NOTES
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached
2. Unless otherwise noted CL = 300 pF VIN = 3V

TIMING CONSIDERATIONS (Commercial Part)

As shown in the "AC CHARACTERISTICS" table for this part, the minimum instruction cycle time is 25 ns, whereas, the maximum is determined by the on-chip oscillator frequency and can be any value the user chooses. With an instruction cycle time of 250 ns, the part can be characterized in terms of absolute values; these are shown in the first "LIMITS" column of the table. When the instruction cycle time is greater than 250 ns, certain parameters are cycle-time dependent; thus, these parameters are specified in terms of the four quarter cycles (T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q}) that make up one instruction cycle — see 8X300 TIMING DIAGRAM. As the time interval for each instruction cycle increases (becomes greater than 250 ns), the delay for all parameters that are cycle-time dependent is likewise increased. In some cases, these delays have a significant impact on timing relationships and other areas of systems design; subsequent paragraphs describe these timing parameters and reliable methods of calculation.

8X300 TIMING DIAGRAM


TPC
TCP
TCL
TCH
X1
1
2
3
4
TMCH
TMCL
MCLK
TAS
TMAS
TW
ADDRESS (A0-A12)
TIA
TACC
TIS
TIVA
INSTRUCTION (I0-I15)
TIH
TWL
TMWL
SC/WC
TMBS
TBS
TWH
TMWH
TIO
TIOS
TOBS
TOOH
TMOOH
TMIOS
TMOBS
IV0-IV7
OUTPUT
INPUT
TMIOH
TIOH
TMHS
THS
TOD
HALT
THH
TMHH
TOOS
TMOOS
INDICATES CHANGING DATA
INDICATES 3-STATE
INPUT PHASE
OUTPUT PHASE
Instruction reg open
instruction address changes
MCLK high
SC = WC = 0
SC = "1" for I/O address, WC = "1" for I/O data
I/O drivers three-state
I/O drivers active if output data or address to I/O BUS, three-state otherwise
I/O receivers open
I/O bus stable
NOTE
ONE CYCLE (~250 ns)

NOTE
For an instruction cycle time greater than 250 ns, the I/O bus can be stable sometime within the third quarter (T_{3Q}) cycle

Timing parameters for the 8X300 are normally measured with reference to X1 or MCLK; those referenced to MCLK are prefaced with an "M" in the mnemonic—TMAS, TMIH, and so on. To determine the timing relationship between a particular signal, say "A" and MCLK, the user should, at all times, use the value specified in the table—DO NOT calculate the value by adding or subtracting two or more parameters that are referenced to X1. When deriving timing relationships between two signals (A to B, etc.) by adding or subtracting the parameter values, the user must consistently use the same parameter reference—MCLK or X1.

System determinants for the instruction cycle time are:

- Propagation delays within the 8X300
- Access time of Program Storage
- Enable time of the I/O port

Normally, the instruction cycle time is constrained by one or more of the following conditions:

Condition 1—Instruction or MCLK to LB/$\overline{\text{RB}}$ (input phase) plus I/O port access time (TIO) ≤ IV data set-up time (Figure 4a).

Condition 2—Program storage access time (TACC) plus instruction to LB/$\overline{\text{RB}}$ (input phase) plus I/O port access time (TIO) plus IV data (input phase) to address ≤ instruction time (Figure 4b).

Condition 3—Program storage access time plus instruction to address ≤ instruction cycle time (Figure 4c).

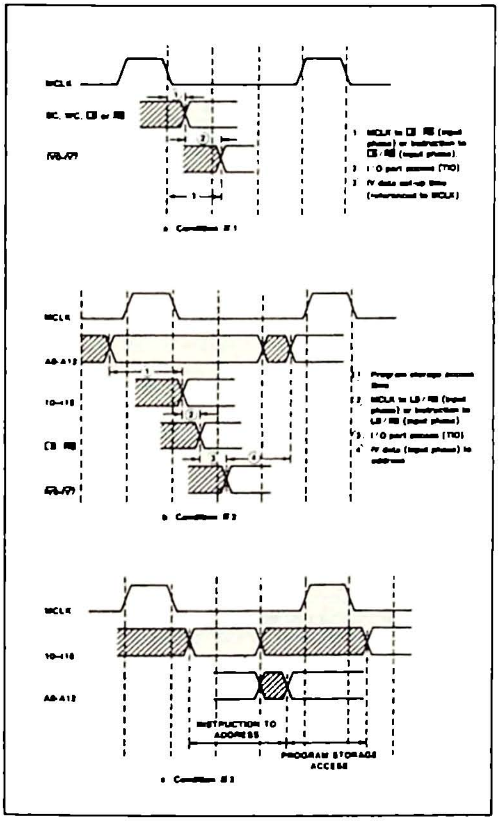

MCLK
SC, WC, LB or RB
IV0–IV7
1
2
3
1 MCLK to LB/RB (input phase) or instruction to LB/RB (input phase)
2 I/O port access (TIO)
3 IV data set-up time (referenced to MCLK)
a. Condition #1
MCLK
A0–A12
I0–I15
LB RB
IV0–IV7
1
2
3
4
1 Program storage access time
2 MCLK to LB/RB (input phase) or instruction to LB/RB (input phase)
3 I/O port access (TIO)
4 IV data (input phase) to address
b. Condition #2
MCLK
I0–I15
A0–A12
INSTRUCTION TO ADDRESS
PROGRAM STORAGE ACCESS
c. Condition #3

Figure 4. Constraints of 8X300 Instruction Cycle Time

From condition #1 and with an instruction cycle time of 250 ns, the I/O port access time (TIO) can be calculated as follows:

TMIBS + TIO ≤ TMIDS
transposing, TIO ≤ TMIDS − TMIBS
substituting, TIO ≤ 55ns − 25ns
result, TIO ≤ 30 ns

Using 30 ns for TIO, the constraint imposed by condition #1 can also be used to calculate the minimum cycle time:

TMIBS + TIO ≤ TMIDS
thus, 25ns + 30ns ≤ $T_{1Q} + T_{2Q}$ − 70
25ns + 30ns ≤ ½ cycle − 70 therefore, the worst-case instruction cycle time is 250 ns. With subject parameters referenced to X1, the same calculations are valid:

TIBS + TIO + TIDS ≤ ½ cycle
thus, 70ns + 30ns + 25ns ≤ ½ cycle therefore, the worst-case instruction cycle time is again 250 ns. From condition #2 and with an instruction cycle time of 250 ns, the program storage access time can be calculated:

TACC + TIIBS + TIO + TIVA ≤ 250ns
transposing, TACC ≤ 250ns − TIIBS − TIO −TIVA
substituting, TACC ≤ 250ns − 35ns - 30ns −105ns
thus, TACC ≤ 80ns hence, for an instruction cycle time of 250 ns, a program storage access time of 80 ns is implied. The constraint imposed by condition #3 can be used to verify the maximum program storage access time

TIA + TACC ≤ Instruction Cycle
thus, TACC ≤ 250ns − 170ns
and, TACC ≤ 80ns, confirming that a program storage access time of 80 ns is satisfactory.

For an instruction cycle time of 250 ns and a program storage access time of 80 ns (Condition #2/Figure 4b), the instruction should be valid 10 ns before the falling edge of MCLK. This relationship can be derived by the following equation:

250ns − TMAS − TACC
= 250ns − 160ns − 80ns
= 10ns

It is important to note that, during the input phase, the beginning of a valid LB/$\overline{\text{RB}}$ signal is determined by either the instruction to LB/$\overline{\text{RB}}$ delay (TIIBS) or the delay from the falling edge of MCLK to LB/$\overline{\text{RB}}$ (TMIBS). Assuming the instruction is valid 10 ns before the falling edge of MCLK and adding the instruction-to-LB/RB delay (TIIBS) = 30ns), the LB/$\overline{\text{RB}}$ signal will be valid 25 ns after the falling edge of MCLK. With a fast program storage memory and with a valid instruction more than 10 ns before the falling edge of MCLK—the LB/$\overline{\text{RB}}$ signal will, due to the TMIBS delay, still be valid 25 ns after the falling edge of MCLK. Using a worst-case instruction cycle time of 250 ns, the user cannot gain a speed advantage by selecting a memory with faster access time. Under the same conditions, a speed advantage cannot be obtained by using an I/O port with fast access time (TIO) because the address bus will be stable 80 ns (TAS) after the beginning of the third quarter cycle—no matter how early the IV data input is valid.

Internal Timing and Timing Relationships

All timing and timing-control signals of the 8X300 are generated by the oscillator and sequencer shown in Figure 5. The sequencer outputs direct and control all of the timing parameters specified in the TIMING DIAGRAM. Observe that each input quarter cycle bears a fixed relationship to X1 via the propagation delay.

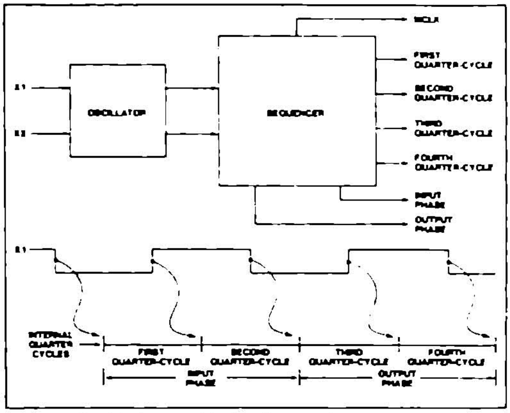

MCLK
FIRST QUARTER-CYCLE
SECOND QUARTER-CYCLE
THIRD QUARTER-CYCLE
FOURTH QUARTER-CYCLE
INPUT PHASE
OUTPUT PHASE
X1
X2
OSCILLATOR
SEQUENCER
X1
INTERNAL QUARTER CYCLES
FIRST QUARTER-CYCLE
SECOND QUARTER-CYCLE
THIRD QUARTER-CYCLE
FOURTH QUARTER-CYCLE
INPUT PHASE
OUTPUT PHASE

Figure 5. Timing and Timing Control Signals of the 8X300

General and interactive timing relationships pertaining to I/O signals of the 8X300 are shown in Figure 6. Example — in the input phase, the switching point of the $\overline{\text{LB}}/\overline{\text{RB}}$ signal is caused by the worst-case delay from the instruction to $\overline{\text{LB}}/\overline{\text{RB}}$ or from the beginning of the first internal quarter cycle to $\overline{\text{LB}}/\overline{\text{RB}}$; the two arrows pointing to the $\overline{\text{LB}}/\overline{\text{RB}}$ transition indicate this "either/or" dependency. This information coupled with tabular values and the TIMING DIAGRAM provides the user with the wherewithal to calculate any and all system timing parameters.

CLOCK CONSIDERATIONS

The on-chip oscillator and timing-generation circuits of the 8X300 can be controlled by any one of the following methods:

Capacitor: if timing is not critical
Crystal: if precise timing is required
External Drive: if application requires that the 8X300 be synchronized with system clock

Capacitor Timing: A non-polarized ceramic or mica capacitor with a working voltage equal to or greater than 25-volts is recommended. The lead lengths of capacitor should be approximately the same and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. For various capacitor (C_X) values, the cycle time can be approximated as:

C_X (In pF)	APPROXIMATE CYCLE TIME
100	300 ns
200	500 ns
500	1.1 μs
1000	2.0 μs

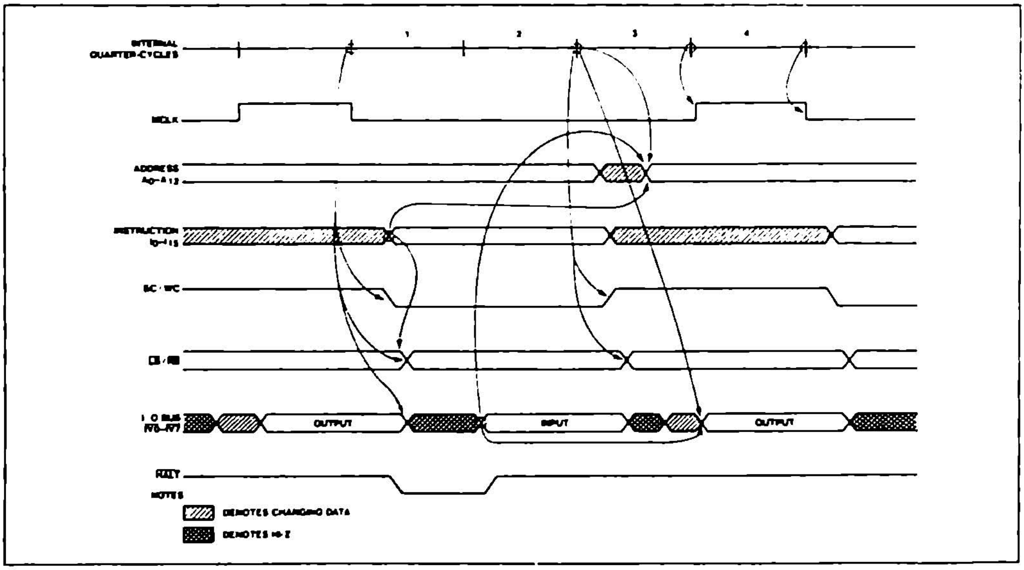

INTERNAL QUARTER-CYCLES
1
2
3
4
MCLK
ADDRESS A0–A12
INSTRUCTION I0–I15
SC/WC
LB/RB
I/O BUS IV0–IV7
OUTPUT
INPUT
OUTPUT
HALT
NOTES
DENOTES CHANGING DATA
DENOTES HI-Z

Figure 6. Timing Relationships of 8X300 I/O Signals

Crystal Timing: When a crystal is used, the on-chip oscillator operates at the resonant frequency (f_O) of the crystal; the series-resonant quartz crystal connects to the 8X300 via pins 10 (X1) and 11 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

Type: Fundamental mode, series resonant
Impedance at Fundamental: 35-ohms maximum
Impedance at Harmonics and Spurs: 50-ohms minimum

The resonant frequency (f_O) of the crystal is related to the desired cycle time (T) by the equation $f_O = 2/T$; for a cycle time of 250 ns, $f_O = 8MHz$.

Using an External Clock: The 8X300 can be synchronized with an external clock by simply connecting appropriate drive circuits to the X1/X2 inputs. Figure 7 shows how the on-chip oscillator can be driven from the complementary outputs of a pulse generator. In applications where the microcontroller must be driven from a master clock, the X1/X2 lines can be interfaced to TTL logic as shown in Figure 8.

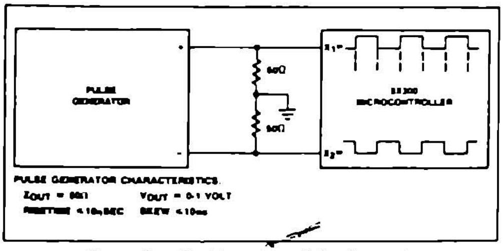


Figure 7. Clocking with a Pulse Generator

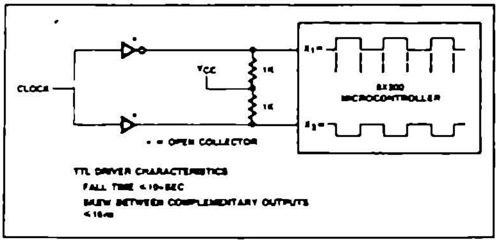


Figure 8. Clocking with TTL

$\overline{\text{RESET}}$ Logic

The $\overline{\text{RESET}}$ line (pin 43) can be driven from a high (inactive) state to a low (active) state at any time with respect to the system clock, that is, the reset function is asynchronous. To ensure proper operation, the $\overline{\text{RESET}}$ line should be held low (active) for one full instruction time. When the line is driven from a high state to an active-low state, several events occur—the precise instant of occurrence is basically a function of the propagation delay for that particular event. As shown in the accompanying $\overline{\text{RESET}}$ timing diagram, these events are:

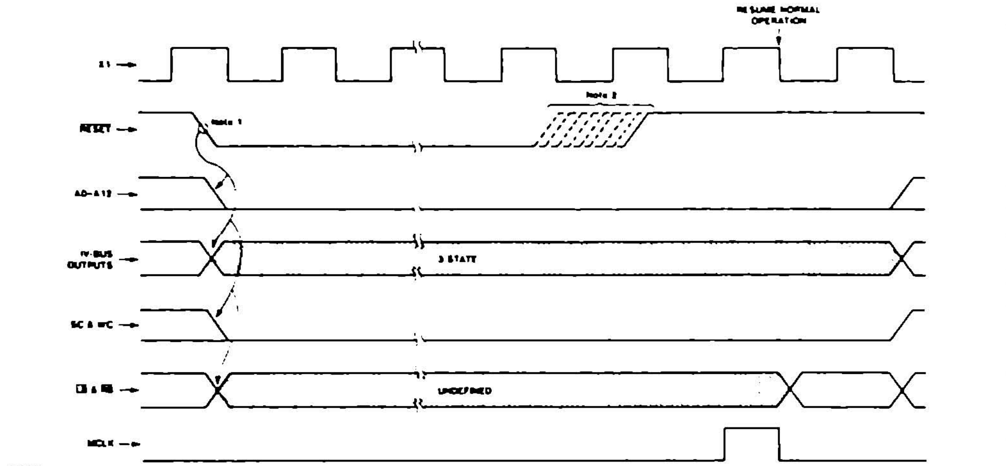
- **The Program Counter and Address Register are set to an all-zero configuration and remain in that state as long as the $\overline{\text{RESET}}$ line is low. Other than PC and AR, reset does not affect other internal registers.**
- **The Input/output (IV) bus goes three-state and remains in that mode as long as the $\overline{\text{RESET}}$ line is low.**
- **The Select Command and Write Command signals are driven low and remain inactive as long as the $\overline{\text{RESET}}$ line is low.**
- **The Left Bank/Right Bank signals are undefined for the period in which the $\overline{\text{RESET}}$ line is low.**

During the time $\overline{\text{RESET}}$ is active-low, MCLK is inhibited; moreover, if the $\overline{\text{RESET}}$ line is driven low during the last two quarter cycles, MCLK can be shortened for that particular machine cycle. When $\overline{\text{RESET}}$ line is driven high (inactive)—one-quarter to one full instruction cycle later—MCLK appears just before normal operation is resumed. The $\overline{\text{RESET}}$/MCLK relationship is clearly shown by "B" in the timing diagram. As long as the $\overline{\text{RESET}}$ line is active-low, the $\overline{\text{HALT}}$ signal (described next) is not sampled by internal logic of the 8X300.

HALT Logic

The $\overline{\text{HALT}}$ signal is sampled via internal chip logic at the end of the first internal quarter of each instruction cycle. If, when sampled, the HALT signal is active-low, a halt is immediately executed and the current instruction cycle is terminated; however, the halt cycle does not inhibit MCLK nor does it affect any internal registers of the 8X300. As long as the $\overline{\text{HALT}}$ line is active-low, the SC and WC lines are low (inactive) and the input/output (IV) bus remains in the three-state mode of operation. The halt cycle continues until, when again sampled, the $\overline{\text{HALT}}$ line is found to be high; at this time, normal operation is resumed. Timing for the halt signal is shown in the accompanying diagram.

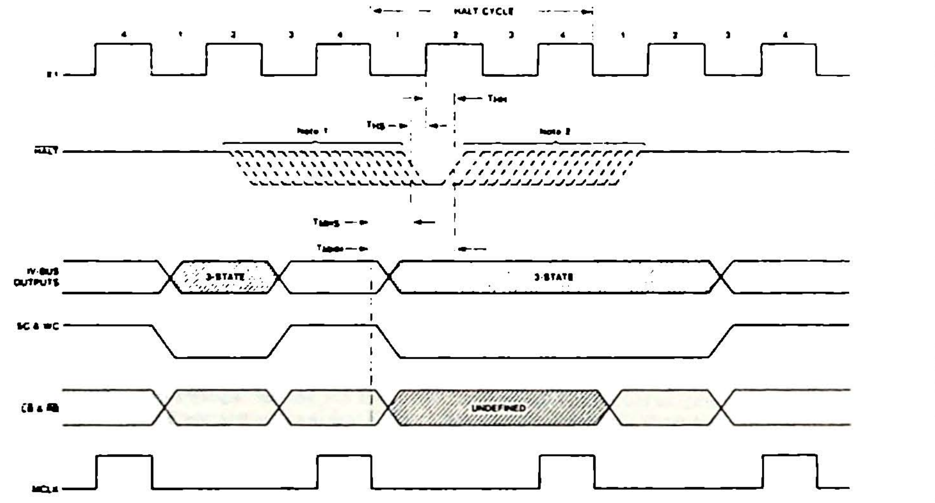
$\overline{\text{RESET}}$ TIMING DIAGRAM


RESUME NORMAL OPERATION
X1
$\overline{\text{RESET}}$
Note 1
Note 2
A0–A12
IV-BUS OUTPUTS
3-STATE
SC & WC
LB & RB
UNDEFINED
MCLK

NOTES

1. A High to Low transition of the $\overline{\text{RESET}}$ signal will force the Address Bus to an all-zero configuration
2. The $\overline{\text{RESET}}$ signal can switch from Low to High at any point within this time interval and in all cases, MCLK will occur at least one-quarter cycle time later as shown

$\overline{\text{HALT}}$ TIMING DIAGRAM


HALT CYCLE
4
1
2
3
4
1
2
3
4
1
2
3
4
X1
T_{HH}
T_{HS}
Note 1
Note 2
$\overline{\text{HALT}}$
T_{MHS}
T_{MHH}
IV-BUS OUTPUTS
3-STATE
3-STATE
SC & WC
LB & RB
UNDEFINED
MCLK

NOTES

1. The $\overline{\text{HALT}}$ signal can switch from High to Low at any time during this interval.
2. The $\overline{\text{HALT}}$ signal can switch from Low to High at any time during this interval

Timing Descriptions:

T_{HS} — set-up time from $\overline{\text{HALT}}$ to X1 (independent of instruction cycle time)

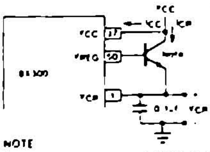
T_{HH} — hold time from X1 to $\overline{\text{HALT}}$ (independent of instruction cycle time)

T_{MHS} — set-up time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

T_{MHH} — hold time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

VOLTAGE REGULATOR

All internal logic of the 8X300 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in the accompanying diagram. To minimize lead inductance, the transistor should be as close as possible to the 8X300 package and the emitter should be ac-grounded via a 0.1-microfarad ceramic capacitor.



PARAMETER	CONDITIONS	LIMITS
h_{FE}	V_{CE} = 3V 100mA ≤ I_C ≤ 500mA	50
$V_{BE(ON)}$	V_{CE} = 3V I_C = 500mA	1V
$V_{CE(SAT)}$	I_C = 500mA I_B = 50mA	0.5V
BV_{CEO}		15V
f_t		30MHz

NOTE
Typical approved parts — 2N5320, 2N5337

JANUARY 1982

MICROCONTROLLER 8X305

FEATURES

- **Fetch, Decode, and Execute a 16-bit Instruction in a minimum of 200 nanoseconds (one machine cycle)**
- **Bit-oriented Instruction set (addressable single-or-multiple bit subfields)**
- **Separate buses for Instruction, Instruction Address and Three-State I/O**
- **Thirteen 8-bit general-purpose working registers**
- **Source/destination architecture**
- **Bipolar low-power Schottky technology/TTL inputs and outputs**
- **On-chip oscillator and timing generation**
- **Single +5V supply**
- **0.9-In. 50-pin DIP**

PRODUCT DESCRIPTION

The Signetics 8X305 MicroController (Figure 1) is a high-speed bipolar microprocessor implemented with low-power Schottky technology. In a single chip, the 8X305 combines speed, flexibility, and a bit-oriented instruction set. These features and other basic characteristics of the chip combine to provide cost-effective solutions for a broad range of applications. The 8X305 is particularly useful in systems that require high-speed bit manipulations — sophisticated controllers, data communications, very fast interface control, and other applications of a similar nature.

The 8X305 can fetch, decode, and execute a 16-bit instruction word in a minimum of 200 nanoseconds. Within one instruction cycle, the 8-bit data-processing path can be programmed to rotate, mask, shift, and/or merge single or multiple bit subfields and, in addition, perform an ALU operation; in the same instruction, an external data field can be input, processed, and output to a specified destination — likewise, single or multiple bit data fields can be internally moved from a given source to a given destination. To summarize, fixed or variable-length data fields can be fetched, processed, operated on by the ALU, and moved to a different location — all in a timeframe of 200 nanoseconds. To interface with I/O and program memory, the 8X305 uses a 13-bit instruction address bus, a 16-bit instruction bus, an 8-bit bidirectional multiplexed I/O data/address bus and a 5-bit I/O control bus.

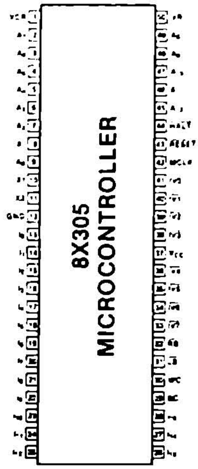
A wide selection of I/O devices, interface chips, and special-purpose parts are available for systems use. In most applications, the more powerful 8X305 is functionally interchangeable with its predecessor — the 8X300.

ASSOCIATED DOCUMENTATION

Other documents directly relating to *design* and *applications use* of the 8X305 MicroController are:

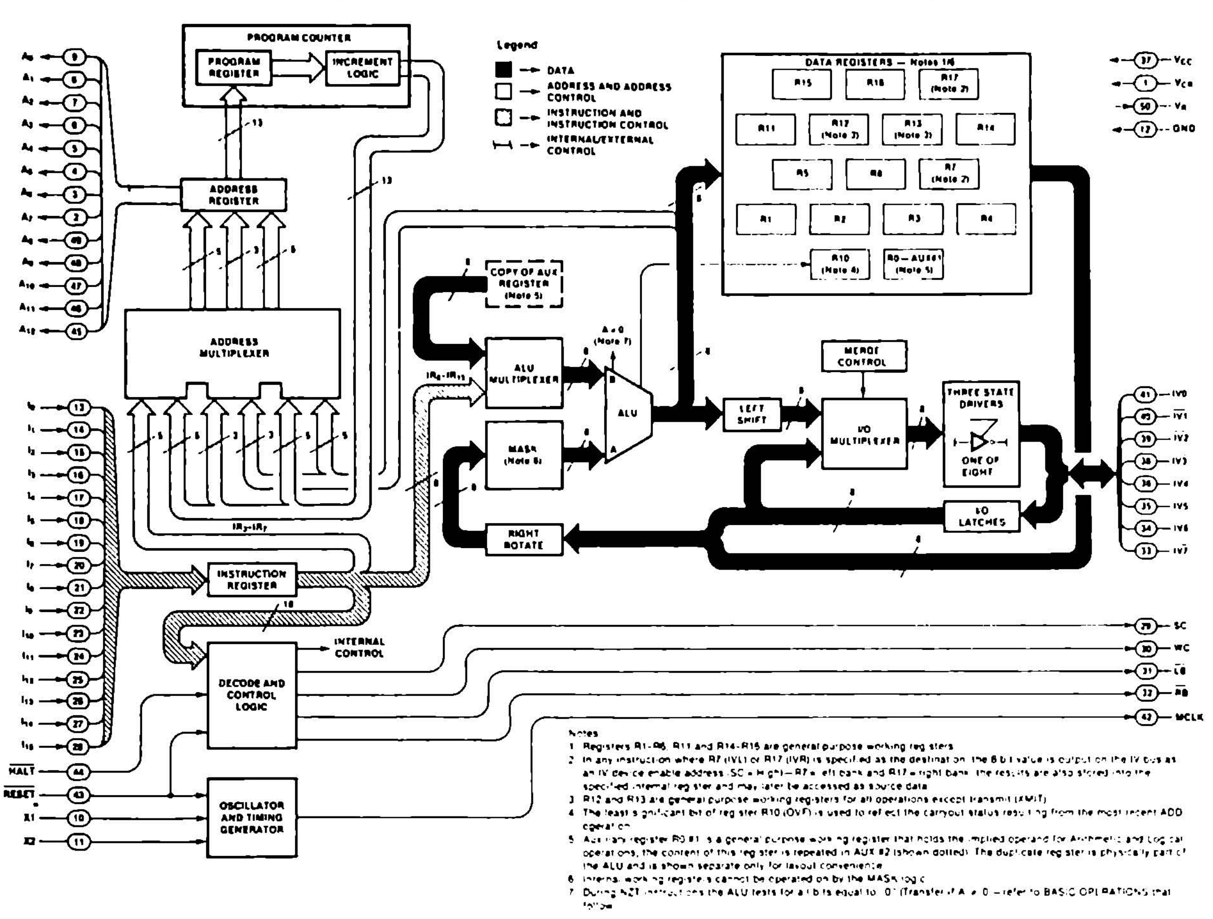
- Product Capabilities Manual for 8X300 FAMILY
- 8X305 Users Manual

These documents and other current literature (Data Sheets, Product Bulletins, Applications Notes, etc.) are available at all Signetics Sales and Service Offices — see rear cover of this data sheet for the office in your locality.



PIN NO.	IDENTIFIER	FUNCTION
1	VCR	Regulated voltage input from series-pass transistor (2N5320 or equivalent).
2-9, 45-49	A_0-A_{12}	**Program Address Lines:** These active-high outputs permit direct addressing of up to 8192 words of program storage; A_{12} is least significant bit.
10, 11	X1, X2	Timing generator connections for a capacitor, a series resonant crystal, or an external clock source with complementary outputs.
12	GND	Ground.
13-28	I_0-I_{15}	**Instruction Lines:** These active-high input lines receive 16-bit instructions from program storage; I_{15} is least significant bit.
29	SC	**Select Command:** When high (binary 1), an address is being output on pins $\overline{IV0}$ through $\overline{IV7}$.
30	WC	**Write Command:** When high (binary 1), data is being output on pins $\overline{IV0}$ through $\overline{IV7}$.
31	$\overline{LB}$	**Left Bank Control:** When low (binary 0), devices connected to the Left Bank are accessed. (Note. *Typically, the $\overline{LB}$ signal is tied to the $\overline{ME}$ input pin of I/O peripherals.)*
32	$\overline{RB}$	**Right Bank Control:** When low (binary 0), devices connected to the Right Bank are accessed (Note. *Typically, the $\overline{RB}$ signal is tied to the $\overline{ME}$ input pin of I/O peripherals.)*
33-36, 38-41	$\overline{IV0}$-$\overline{IV7}$	Interface Vector (Input/Output Bus) — these bidirectional active-low three-state lines communicate data and/or addresses to I/O devices and memory locations. A low voltage level equals a binary "1"; $\overline{IV7}$ is Least Significant Bit.
37	V_{CC}	+ 5V power supply.
42	MCLK	**Master Clock:** This active-high output signal is used for clocking I/O devices and/or synchronization of external logic.
43	$\overline{RESET}$	When $\overline{RESET}$ input is low (binary 0), the 8X305 is initialized — sets Program Counter/Address Register to zero and inhibits MCLK. For the period of time $\overline{RESET}$ is low, the Left Bank/Right Bank ($\overline{LB}$/$\overline{RB}$) signals are forced high asynchronously.
44	$\overline{HALT}$	When $\overline{HALT}$ input is low (binary 0), internal operation of the 8X305 stops at the start of next instruction; MCLK is not inhibited nor is any internal register affected; however, both the Left Bank/ Right Bank ($\overline{LB}$/$\overline{RB}$) signals are synchronously driven high during the first quarter of the instruction cycle time and remain high during the time $\overline{HALT}$ is low.
50	VR	Internally-generated reference output voltage for external series-pass regulator transistor.

Figure 2. Designations and Descriptions for Pins of 8X305 MicroController


PROGRAM COUNTER
PROGRAM REGISTER
INCREMENT LOGIC
ADDRESS REGISTER
ADDRESS MULTIPLEXER
INSTRUCTION REGISTER
DECODE AND CONTROL LOGIC
INTERNAL CONTROL
OSCILLATOR AND TIMING GENERATOR
Legend
DATA
ADDRESS AND ADDRESS CONTROL
INSTRUCTION AND INSTRUCTION CONTROL
INTERNAL/EXTERNAL CONTROL
DATA REGISTERS — Notes 1/6
R15
R16
R17 (Note 2)
R11
R12 (Note 3)
R13 (Note 3)
R14
R5
R6
R7 (Note 2)
R1
R2
R3
R4
R10 (Note 4)
R0—AUX#1 (Note 5)
COPY OF AUX REGISTER (Note 5)
ALU MULTIPLEXER
MASK (Note 6)
ALU
A = 0 (Note 7)
B
A
LEFT SHIFT
MERGE CONTROL
I/O MULTIPLEXER
THREE STATE DRIVERS
ONE OF EIGHT
I/O LATCHES
RIGHT ROTATE
IR8-IR12
IR0-IR7
13
8
5
3
A0 9
A1 8
A2 7
A3 6
A4 5
A5 4
A6 3
A7 2
A8 49
A9 48
A10 47
A11 46
A12 45
I0 13
I1 14
I2 15
I3 16
I4 17
I5 18
I6 19
I7 20
I8 21
I9 22
I10 23
I11 24
I12 25
I13 26
I14 27
I15 28
HALT 44
RESET 43
X1 10
X2 11
37 VCC
1 VCR
50 VR
12 GND
41 IV0
40 IV1
39 IV2
38 IV3
36 IV4
35 IV5
34 IV6
33 IV7
29 SC
30 WC
31 LB
32 RB
42 MCLK

Notes

1. Registers R1-R6, R11 and R14-R15 are general purpose working registers.
2. In any instruction where R7 (IVL) or R17 (IVR) is specified as the destination, the 8-bit value is output on the IV bus as an IV device enable address (SC = High)—R7 = left bank and R17 = right bank. The results are also stored into the specified internal register and may later be accessed as source data.
3. R12 and R13 are general purpose working registers for all operations except transmit (XMIT).
4. The least significant bit of register R10 (OVF) is used to reflect the carryout status resulting from the most recent ADD operation.
5. Auxiliary register R0 #1 is a general purpose working register that holds the implied operand for Arithmetic and Logical operations; the content of this register is repeated in AUX #2 (shown dotted). The duplicate register is physically part of the ALU and is shown separate only for layout convenience.
6. Internal working registers cannot be operated on by the MASK logic.
7. During NZT instructions the ALU tests for all bits equal to "0" (Transfer if A ≠ 0 — refer to BASIC OPERATIONS that follow).

Figure 1. Architecture and Pin Designations for 8X305 MicroController

FUNCTIONAL OPERATION

Typical System Configuration

Although the system hookup shown in Figure 3 is of the simplest form, it provides a fundamental look at the 8X305 MicroController and peripheral relationships. As indicated, the 8X305 can directly address up to 8K words of program storage — either ROM or PROM. The user interface ($\overline{IV0}$ through $\overline{IV7}$) is capable of uniquely addressing 256 Input/Output locations and, with additional bank bits ($\overline{LB}$, $\overline{RB}$), this number is expanded to 512 — each bank comprising 256 addressable locations. The addressable locations of each bank can be used in a variety of ways, a simple method of implementation is shown in Figure 3. When $\overline{LB}$ is active low, the left bank is enabled and anyone of 256 locations within the RAM memory can be accessed for input/output operations. A similar set of "enable/access" conditions are applicable to the right bank when $\overline{RB}$ is active low.

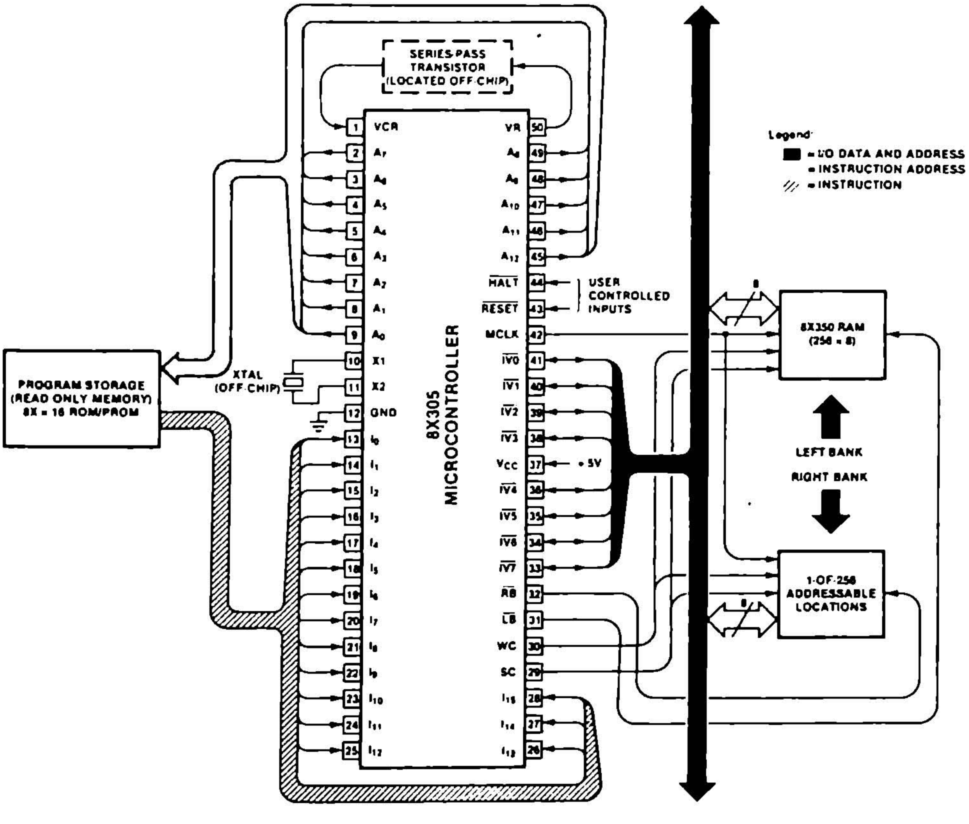

SERIES-PASS
TRANSISTOR
(LOCATED OFF-CHIP)
VCR
VR
A7
A6
A5
A4
A3
A2
A1
A0
A8
A9
A10
A11
A12
HALT
RESET
USER CONTROLLED INPUTS
MCLK
X1
X2
GND
XTAL (OFF-CHIP)
PROGRAM STORAGE (READ ONLY MEMORY) 8K × 16 ROM/PROM
8X305 MICROCONTROLLER
IV0
IV1
IV2
IV3
Vcc
+5V
IV4
IV5
IV6
IV7
RB
LB
WC
SC
I0
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
Legend:
■ = I/O DATA AND ADDRESS
— = INSTRUCTION ADDRESS
/// = INSTRUCTION
8
8X350 RAM (256 × 8)
LEFT BANK
RIGHT BANK
1-OF-256 ADDRESSABLE LOCATIONS

Figure 3. Typical 8X305 System Hookup

BASIC OPERATIONS OF 8X305

Refer to a later discussion of "Instruction Fields" for a detailed examination of all operand fields and subdivisions thereof—"S" (S_0, S_1), "D" (D_0, D_1), "R", "L", "J", and "A"

MOVE OPERATIONS

REGISTER-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0–R17 as specified by "S" field of instruction	Right rotate as specified by "R" field of instruction	No operation	No operation	R0–R7, R11–R17 as specified by "D" field of instruction

REGISTER-TO-$\overline{\text{IV}}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0–R17 as specified by "S" field of instruction.	No operation	No operation	Shift and merge as specified by "D_0" and "L" fields of instruction	Variable length field of $\overline{\text{IV}}$ bus—Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "D_0" and "D_1" fields of instruction

$\overline{\text{IV}}$ BUS-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{\text{IV}}$ bus—Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "S_1" and "S_0" fields of instruction.	Right rotate and mask as specified by "S_0" and "L" fields of instruction.	No operation	No operation.	R0–R7, R11–R17 as specified by "D" field of instruction

$\overline{\text{IV}}$ BUS-TO-$\overline{\text{IV}}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{\text{IV}}$ bus—Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "S_1" and "S_0" fields of instruction	Right rotate and mask as specified by "S_0" and "L" fields of instruction.	No operation.	Shift and merge as specified by "D_0" and "L" fields of instruction	Variable length field of $\overline{\text{IV}}$ bus—Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "D_0" and "D_1" fields of instruction

ADD OPERATIONS

REGISTER-to-REGISTER → SOURCE → PRE-ALU

REGISTER-TO-$\overline{\text{IV}}$ BUS → SOURCE → PRE-ALU

$\overline{\text{IV}}$ BUS-to-REGISTER → SOURCE → PRE-ALU

$\overline{\text{IV}}$ BUS-to-$\overline{\text{IV}}$ BUS → SOURCE → PRE-ALU

Same as MOVE operations, except source data is ADDed to contents of AUXiliary Register R0 via the ALU; if appropriate, Overflow Register R10 (OVF) is also set.

POST-ALU → DEST

POST-ALU → DEST

POST-ALU → DEST

POST-ALU → DEST

AND OPERATIONS

REGISTER-to-REGISTER → SOURCE → PRE-ALU

REGISTER-TO-$\overline{\text{IV}}$ BUS → SOURCE → PRE-ALU

$\overline{\text{IV}}$ BUS-to-REGISTER → SOURCE → PRE-ALU

$\overline{\text{IV}}$ BUS-to-$\overline{\text{IV}}$ BUS → SOURCE → PRE-ALU

Same as MOVE operations, except source data is ANDed with contents of AUXiliary Register R0 via the ALU.

POST-ALU → DEST

POST-ALU → DEST

POST-ALU → DEST

POST-ALU → DEST

EXCLUSIVE OR (XOR) OPERATIONS

REGISTER-to-REGISTER → SOURCE → PRE-ALU

REGISTER-TO-$\overline{\text{IV}}$ BUS → SOURCE → PRE-ALU

$\overline{\text{IV}}$ BUS-to-REGISTER → SOURCE → PRE-ALU

$\overline{\text{IV}}$ BUS-to-$\overline{\text{IV}}$ BUS → SOURCE → PRE-ALU

Same as MOVE operations, except source data is exclusively ORed with contents of AUXiliary Register R0 via the ALU.

POST-ALU → DEST

POST-ALU → DEST

POST-ALU → DEST

POST-ALU → DEST

EXECUTE (XEC) OPERATIONS

XEC, REGISTER

SOURCE	DATA PROCESSING (PRE ALU)	ALU	DATA PROCESSING (POST ALU)	DESTINATION
R0–R17 as specified by "S" field of instruction	No operation	Add source data to 8 bit field specified by instruction literal (0 ≤ J ≤ 377_8)	No operation	Replace 8 LSB of Address Register* with 8 bit sum from ALU *PGM CTR unchanged

XEC, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction	Rotate and mask as specified by "S_0" and "L" fields of instruction	Add masked field of source data to 5-bit literal specified by "J" field of instruction (0 ≤ J ≤ 37_8)	No operation	Replace 5 LSB of Address Register* with 5-bit sum from ALU *PGM CTR unchanged

NON-ZERO TRANSFER (NZT) OPERATIONS

NZT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU		DESTINATION
R0–R17 as specified by "S" field of instruction	No operation	Test contents of source register for all zeroes		If S = 0, increment PC by 1, if S ≠ 0, replace 8 LSB of AR and PC with literal specified by "J" field of instruction

NZT, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU		DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction	Rotate and mask as specified by "S_0" and "L" fields of instruction	Test contents of masked field for all zeroes		If S = 0, increment PC by 1, if S ≠ 0, replace 5 LSB of AR and PC with literal specified by "J" field of instruction

TRANSMIT (XMIT) OPERATIONS

XMIT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
0 ≤ J ≤ 377_8 – value specified by "J" field of instruction	No operation	No operation	No operation	R0–R7, R11, R14–R17. Load 8 bit integer specified by "J" field into register specified by "D" field

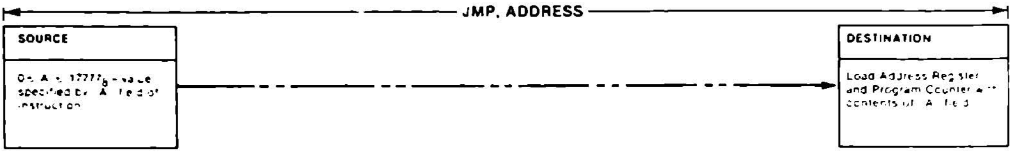
XMIT 8-BITS IMMEDIATE, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
0 ≤ J ≤ 377_8 – value specified by "J" field of instruction.	No operation	No operation	No operation	Left (R12) or Right (R13) Bank of $\overline{IV}$ bus as specified by "D" field of instruction

XMIT VARIABLE-BIT FIELD IMMEDIATE $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
0 ≤ J ≤ 37_8 – value specified by "J" field of instruction	No operation	No operation	Shift and merge source data as specified by "D_0" and "L" fields of instruction	Left or Right Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

JUMP (JMP) OPERATION


JMP. ADDRESS
SOURCE
On A = 17777₈ - value specified by A field of instruction
DESTINATION
Load Address Register and Program Counter with contents of A field

Program Storage Interface

As shown in Figure 3, program storage is connected to output address lines A_0 through A_{12} (A_{12} = LSB) and input instruction lines I_0 through I_{15}. An address output on A_0/A_{12} identifies one 16-bit instruction word in program storage. The instruction word is subsequently input on I_0/I_{15} and defines the MicroController operation which is to follow — one instruction word equals one completed operation. Any TTL-compatible memory can be used for program storage provided the worst-case access time is compatible with the instruction cycle time used for the application — see timing section for appropriate calculations.

I/O Interface and Control

An 8-bit bidirectional I/O bus, referred to as the Interface Vector ($\overline{IV}$) bus, provides a communication link between the MicroController and the two banks of I/O devices. The $\overline{LB}$ (Left Bank) and $\overline{RB}$ (Right Bank) control signals identify which bank is enabled; when both $\overline{LB}$ and $\overline{RB}$ are high (inactive), neither bank is enabled and the $\overline{IV}$ bus is inactive (three-state). A functional analysis of the Left and Right Bank signals is shown below:

$\overline{LB}$	$\overline{RB}$	FUNCTION
Low	Low	This state is not generated by the 8X305
Low	High	Enable left bank devices
High	Low	Enable right bank devices.
High	High	Disable all devices; $\overline{IV}$ bus is three-state.

Both data and I/O address information are multiplexed on the $\overline{IV}$ bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and I/O address information as follows:

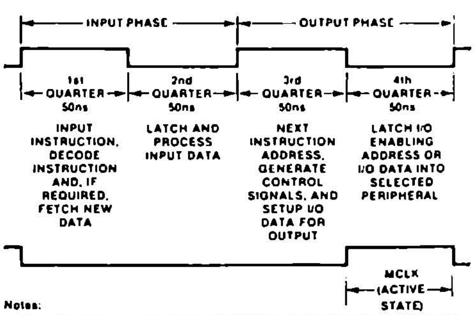
*SC	WC	FUNCTION
Low	Low	Input data expected from selected I/O device. *(Note. This expectation is not true for internal operations or during the output phase of instruction cycle.)*
Low	High	Data is being output on the $\overline{IV}$ bus.
High	Low	Address is being output on the $\overline{IV}$ bus.
High	High	This state is not generated by the 8X305.

Data Processing

Basically, the data processing path of the 8X305 consists of the Rotate/Mask logic, the Arithmetic Logic Unit (ALU), the Shift/ Merge functions, on-chip memory (sixteen 8-bit registers), and the bidirectional $\overline{IV}$ bus interface with its associated driver circuits and internal latches. The on-board memory and the $\overline{IV}$ bus are connected to both inputs and outputs of the ALU via internal 8-bit data paths — see Figure 1. Inputs to the ALU are preceded by right-rotate and data-mask functions; the ALU output is followed by the left-shift and merge operations. Depending on the desired operation, any one or all of the functions (Rotate/Mask/Shift/Merge) can operate on 8 bits of data in a single instruction cycle. For a summary of all data-processing capabilities, refer to BASIC OPERATIONS OF THE 8X305 described earlier in this data sheet.

Instruction Cycle

Each operation of the 8X305 is executed in a single instruction cycle. The instruction cycle is internally divided into four equal parts — each part being as short as 50 nanoseconds. Figure 4 shows the general functions that


INPUT PHASE
OUTPUT PHASE
1st QUARTER 50ns
2nd QUARTER 50ns
3rd QUARTER 50ns
4th QUARTER 50ns
INPUT INSTRUCTION, DECODE INSTRUCTION AND, IF REQUIRED, FETCH NEW DATA
LATCH AND PROCESS INPUT DATA
NEXT INSTRUCTION ADDRESS, GENERATE CONTROL SIGNALS, AND SETUP I/O DATA FOR OUTPUT
LATCH I/O ENABLING ADDRESS OR I/O DATA INTO SELECTED PERIPHERAL
MCLK (ACTIVE STATE)

Notes:

1. New instruction must be accepted and latched at end of first quarter cycle
2. The I/O data latches are open for the first two quarter cycles, that is, for 100 nanoseconds
3. The address is output during third quarter cycle
4. $\overline{IV}$ bus drivers are active (turned on) during third and fourth quarter cycles

Figure 4. Instruction Cycle and MCLK with: Crystal = 10MHz and Cycle Time = 200 nanoseconds.

occur during each quarter cycle; specifics regarding minimum/maximum timing and other critical values are described later in this data sheet During the first quarter cycle, a new instruction from program storage is input via I_0-I_{15} and decoded. If an I/O operation is indicated, new data is fetched from a specified internal register or via the $\overline{IV}$ bus. At the end of the first quarter cycle, the new instruction is latched into the instruction register

In the second quarter cycle, the I/O input data stabilizes and preliminary processing is completed; at the end of this quarter, the $\overline{IV}$ latches close and final processing can be accomplished, thus completing the input phase of the instruction cycle. During the third quarter cycle, the address for the next instruction is output to the instruction address bus, $\overline{IV}$ control signals are generated, and both data and destination are setup for the remainder of the output phase. During the fourth quarter cycle, a master clock signal (MCLK) generated by the 8X305 is used to latch either the I/O-enabling address or the I/O data into peripheral devices connected to the $\overline{IV}$ bus; MCLK can also be used to synchronize any external logic with timing circuits of the 8X305. To summarize the action, the first half of the instruction cycle deals primarily with input functions and the second half is mostly concerned with output functions.

INSTRUCTION SET

General Format and Operating Principles

The 16-bit instruction word (I_0 through I_{15}) from program storage is input to the instruction register (Figure 1) and is subsequently decoded to implement the events to occur during the current instruction cycle.

The general format for each instruction word is as follows:

↓ MSB LSB ↓

BIT POSITIONS →	0 1 2	3 4 5 6 7 8 9 10 11 12 13 14 15
	OPCODE	OPERAND(S)

The 3-bit operation code (OPCODE) define any one of eight classes of instructions; variations within each class are specified by the remaining thirteen operand bits. The eight instruction classes can be separated into two control areas — *data* and *program;* general functions within these areas are:

- Data Control —
  - ADD, AND, XOR } Arithmetic and Logic Operations
  - MOVE, XMIT } Movement of Data and Constants
- Program Control
  - XEC, NZT, JMP } Branch or Test

Instruction Fields

As shown in Table 1, each instruction word consists of an operation code (OPCODE) field and from one to three operand fields. The possible operand fields are: Source (S), Destination (D), Rotate/Length (R/L), Literal (J), and Address (A). The OPCODE and operand fields are described in the paragraphs that follow the table.

Table 1. Functional Description of Instruction Set

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 4		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = MOVE OPCODE = 0 OPERATION = (S) → D				
Register-to-Register 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S R D S = 20_8-37_8 D = 10_8, 20_8-37_8	Move content of internal register specified by S field to internal register specified by D field. Prior to the MOVE operation right rotate contents of internal source register by octal value (0 through 7) defined by the R field	SC	L	H if D = 07_8 17_8
		WC	L	L
		$\overline{LB}$	H	L if D = 07_8
		$\overline{RB}$	H	L if D = 17_8
Register-to-$\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L D D_1 : D_0 S = 20_8-37_8 D = 20_8-37_8	Move contents of internal register specified by the S field to the $\overline{IV}$ bus. Before outputting on $\overline{IV}$ bus data is shifted as specified by the least significant octal digit of the D field and the bits specified by the L field are merged with the latched I/O data	SC	L	L
		WC	L	H
		$\overline{LB}$	L if D = 20_8-27_8	L if D = 20_8-27_8
		$\overline{RB}$	L if D = 30_8-37_8	L if D = 30_8-37_8

Table 1. Functional Description of Instruction Set (Continued)

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 3		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = MOVE OPCODE = 0 OPERATION = (S) → D				
$\overline{IV}$ Bus-to-Register (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L D S_1 : S_0 $S = 20_8$-37_8 $D = 10_8$, 20_8-37_8	Move right-rotated $\overline{IV}$ bus (source) data specified by the S-field to internal register specified by the D-field. The L-field specifies the length of source data starting from the LSB position and, if less than 8 bits, the remaining bits are filled with zeros.	SC	L	H if $D = 07_8$, 17_8
		WC	L	L
		$\overline{LB}$	L if $D = 20_8$-27_8	L if $D = 07_8$
		$\overline{RB}$	L if $D = 30_8$-37_8	L if $D = 17_8$
$\overline{IV}$ Bus-to-$\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L D S_1 : S_0 D_1 D_0 $S = 20_8$-37_8 $D = 20_8$-37_8	Move right-rotated $\overline{IV}$ bus (source) data specified by the S-field to the I/O latches. Before outputting on $\overline{IV}$ bus, shift data as specified by the D field, then merge source and latched I/O data as specified by the L (length) field.	SC	L	L
		WC	L	H
		$\overline{LB}$	L if $S = 20_8$-27_8	L if $D = 20_8$-27_8
		$\overline{RB}$	L if $S = 30_8$-37_8	L if $D = 30_8$-37_8
CLASS = ADD OPCODE = 1 OPERATION = (S) + (AUX) → D				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ADDed to the source data. If there is a "carry" from MSB, then R10 (OVF) = 1 (overflow), otherwise OVF = 0	Same as MOVE instruction class		
CLASS = AND OPCODE = 2 OPERATION = (S) ∧ (AUX) → D				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ANDed with source data	Same as MOVE instruction class		
CLASS = XOR OPCODE = 3 OPERATION = (S) ⊕ (AUX) → D				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are exclusively ORed with source data.	Same as MOVE instruction class		
CLASS = XEC OPCODE = 4 OPERATION = Refer to Description				
Register Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S J $S = 20_8$-37_8 $J = 000_8$-377_8	Execute instruction at current page address offset by J (literal) + (S). Return to normal instruction flow unless a branch is encountered. Execute instruction at an address determined by replacing the low-order 8 bits of the Address Register with the following derived sum: Value of literal (J-field) plus contents of internal register specified by S-field. The PC is not incremented and the overflow status (OVF) is not changed.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H
$\overline{IV}$ Bus Immediate (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L J S_1 : S_0 $S = 20_8$-37_8 $J = 00_8$-37_8	Execute instruction at an address determined by replacing the low-order 5 bits of Address Register with the following derived sum: 5-bit value of literal (J-field) plus value of rotated source data specified by S-field. The L-field specifies the length of source data starting from the LSB position and, if less than 8 bits, the remaining bits are filled with zeros; the Program Counter is not incremented and the overflow status (OVF) is not changed.	SC	L	L
		WC	L	L
		$\overline{LB}$	L if $S = 20_8$-27_8	H
		$\overline{RB}$	L if $S = 30_8$-37_8	H
CLASS = NZT OPCODE = 5 OPERATION = Refer to Description				
Register Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S J $S = 20_8$-37_8 $J = 000_8$-377_8	If data specified by the S-field is not equal to zero, jump to current page address offset by value of J-field; otherwise, increment the Program Counter. If contents of internal register specified by S-field is non-zero, transfer to address determined by replacing the low-order 8 bits of Address Register and Program Counter with "J", otherwise, increment PC.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H

Table 1 Functional Description of Instruction Set (Concluded)

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 3		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = NZT OPCODE = 5 OPERATION = Refer to Description				
$\overline{IV}$ Bus Immediate (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE · S (S_1 : S_0) · L · J $S = 20_8 - 37_8$ $J = 00_8 - 37_8$	If right rotated and masked $\overline{IV}$ bus is non-zero transfer to address determined by replacing low-order 5 bits of Address Register and Program Counter with "J", otherwise increment PC. (The "L" field specifies the length of source I/O data starting from the LSB position and, if less than 8 bits the remaining bits are filled with zeros.)	SC	L	L
		WC	L	L
		$\overline{LB}$	L if $S = 20_8 - 27_8$	H
		$\overline{PB}$	L if $S = 30_8 - 37_8$	H
CLASS = XMIT OPCODE = 6 OPERATION = J → D				
XMIT, Register 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE · D · J $D = 10_8, 12_8, 13_8, 07_8, 17_8, 20_8 - 37_8$ $J = 000_8 - 377_8$	Store 8-bit value specified by "J" into register specified by "D".	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{PB}$	H	H
XMIT, $\overline{IV}$ Bus Address 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE · D · J $D = 07_8, 17_8$ $J = 000_8 - 377_8$	Enable I/O device on the bank specified by "D" whose address is the 8-bit integer specified by "J". Address "J" is stored in register "D".	SC	L	H
		WC	L	L
		$\overline{LB}$	H	L if $D = 07_8$
		$\overline{PB}$	H	L if $D = 17_8$
XMIT 8 Bits Immediate, $\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE · D · J $D = 12_8 - 13_8$ $J = 000_8 - 377_8$	Store value of 8-bit integer in the previously enabled I/O port, at the bank destination ($\overline{LB}$ or $\overline{PB}$) specified by "D". Contents of R12 or R13 remain unchanged.	SC	L	L
		WC	L	H
		$\overline{LB}$	H	L if $D = 12_8$
		$\overline{PB}$	H	L if $D = 13_8$
XMIT Variable Bit Field Immediate, $\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE · D (D_1 : D_0) · L · J $D = 20_8 - 37_8$ $J = 00_8 - 37_8$	Transmit Least Significant "L" bits of "J" field to "L bit" field of $\overline{IV}$ bus specified by "D"; if "L" is greater than 5 bits, the MSB bits of destination field is filled with zeroes.	SC	L	L
		WC	L	H
		$\overline{LB}$	L if $D = 20_8 - 27_8$	L if $D = 20_8 - 27_8$
		$\overline{PB}$	L if $D = 30_8 - 37_8$	L if $D = 30_8 - 37_8$
CLASS = JMP OPCODE = 7 OPERATION = Refer to Description				
Address Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE · A $A = 0000_8 - 17777_8$	Jump to address in program storage specified by A field. This address is loaded into the Address Register and the Program Counter.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{PB}$	H	H

Note

S_0 specifies the LSB of rotated input data field

S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input

D_0 specifies bit position in I/O device with which LSB of processed data will be aligned and

D_1 specifies the bank of $\overline{IV}$ bus which will be the destination

Operations Code Field. The 3-bit OPCODE field specifies one of eight classes of 8X305 instructions; octal designations for this field and operands for each instruction class are shown in the preceding table.

Source (S) and Destination (D) Fields. The 5-bit "S" and "D" fields specify the source and destination, respectively, for whatever operation is defined by the OPeration CODE. The "S" and/or "D" fields can specify an internal 8X305 register or any one-to-eight bit field within an I/O device; octal values and source/destination field assignments for all internal registers are shown in Table 2.

Table 2. Octal Addresses and Source/Destination Fields for 8X305 Registers

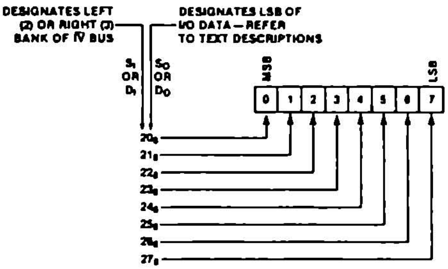
ADDRESS	REGISTER DESIGNATION	SOURCE	DESTINATION	ADDRESS	REGISTER DESIGNATION	SOURCE	DESTINATION
00_8	R0 (AUX)—General purpose register	X	X	10_8	R10 (OVF—Overflow register)	X	
01_8	R1—General purpose register	X	X	11_8	R11—General purpose register	X	X
02_8	R2—General purpose register	X	X	12_8	R12—General purpose register (Note)	X	X
03_8	R3—General purpose register	X	X	13_8	R13—General purpose register (Note)	X	X
04_8	R4—General purpose register	X	X	14_8	R14—General purpose register	X	X
05_8	R5—General purpose register	X	X	15_8	R15—General purpose register	X	X
06_8	R6—General purpose register	X	X	16_8	R16—General purpose register	X	X
07_8	R7—Special purpose register (refer to next paragraph)	X	X	17_8	R17—Special purpose register (refer to next paragraph)	X	X

Note:

R12 and R13 function as general purpose working registers for all operations except transmit (XMIT). During a transmit instruction where R12 or R13 is the destination, the 8-bit "J" field is immediately transferred to the $\overline{\text{IV}}$ bus; for this operation, the contents of the designated register remain unchanged.

In instructions where $R7_8$ (IVL) or $R17_8$ (IVR) is specified as the destination, the 8-bit value is output on the $\overline{\text{IV}}$ bus as an I/O device address or memory location; register R7 selects the Left Bank and register R17 selects the Right Bank. The results are also stored into the specified internal register ($R7_8$ or $R17_8$) and may later be accessed as source data. When the $\overline{\text{IV}}$ bus is specified as a source and/or destination, the "S" and "D" fields are split into two parts, that is,

- Source (S) = S_1, S_0 and Destination (D) = D_1, D_0 where,
  S_0 specifies the LSB of rotated input data field
  S_1 specifies the bank of $\overline{\text{IV}}$ bus from which source data will be input
  D_0 specifies bit position in I/O device with which LSB of processed data will be aligned and
  D_1 specifies the bank of $\overline{\text{IV}}$ bus which will be the destination


DESIGNATES LEFT (2) OR RIGHT (3) BANK OF $\overline{\text{IV}}$ BUS
DESIGNATES LSB OF I/O DATA—REFER TO TEXT DESCRIPTIONS
S_1 OR D_1
S_0 OR D_0
MSB
LSB
0
1
2
3
4
5
6
7
20_8
21_8
22_8
23_8
24_8
25_8
26_8
27_8

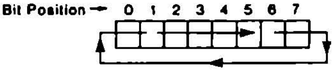
Notes

1. The field length of 0-to-8 bits is specified by the "L" field
2. For the Right Bank, 30_8-37_8 perform equivalent I/O functions

Rotate (R) and Length (L) Field. The 3-bit R/L field performs one of two functions, specifying either the field length (L) for I/O operations or a right-rotate (R) for internal operations. For a given instruction, the specified function depends upon the contents of the Source (S) and Destination (D) fields.

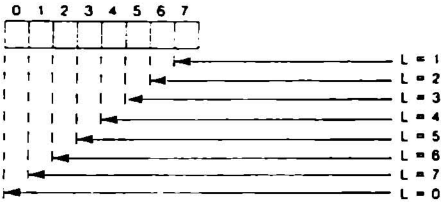
When an internal register is specified by both the source and destination fields, the "R" field is invoked and it specifies a right-rotate of the data specified in the "S" field — see accompanying diagram. The source-register data (up to 8 bits) is right-rotated during the "input phase" of the instruction cycle (Figure 4) and this function is always performed prior to any ALU operation. (*Note: The right-rotate function is implemented on the bus and not in the source register.*)

RIGHT-ROTATE FUNCTION



When either or both of the source and destination fields specify a variable-length I/O data field, the "L" field specifies the length of the I/O data field — see following diagram. If the source field specifies an $\overline{\text{IV}}$ address (20_8-37_8) and the destination field specifies an internal register (00_8-07_8, 11_8-17_8), the "L" field specifies the length of source data; the source data is formed by right-rotating the $\overline{\text{IV}}$ bus data according to the source address and then masking result as specified by the "L" field. If length is less than 8 bits, all remaining bits are set to zero prior to processing data in the ALU. If the source field specifies an internal register (00_8-17_8) and the destination field specifies $\overline{IV}$ bus data (20_8-37_8), the "L" field specifies the length of the destination data. To form the destination data, the ALU output is left-shifted according to the destination address and then masked to the required length — see $\overline{IV}$ DATA LENGTH SPECIFICATION. The destination data is merged with data in the I/O latches to finalize the $\overline{IV}$ bus data. Hence, a one-to-eight bit destination data field can be inserted into the existing 8-bit I/O port without modifying surrounding bits. If both the source and destination fields specify $\overline{IV}$ bus data (20_8-37_8), the "L" field specifies the length of both the source and destination data.

$\overline{IV}$ DATA LENGTH SPECIFICATION
(No Rotate Function Specified)



To form the source data, the $\overline{IV}$ bus input data is right-rotated according to the source address and then masked to the required length — see $\overline{IV}$ DATA LENGTH SPECIFICATION. If length is less than 8 bits, all remaining bits are set to zero before processing in the ALU. To form the destination data, the ALU output is left-shifted according to the destination address and masked to the required length specification. The destination data is then merged into the $\overline{IV}$ bus data that was used to obtain the source; thus, if the source and destination addresses are on the same bank, the $\overline{IV}$ bus data written to the destination I/O Port appears unmodified, except for bits changed during the shift-and-mask operations. If the source and destination addresses refer to different banks, the destination I/O Port is changed to contain the contents of the source I/O Port in those bit positions not affected by the destination data.

J Field. The 5-bit or 8-bit "J" field is used to load a literal value (contained in the instruction) into a register, into a variable I/O data field, or to modify the low-order bits of the Program Counter. The bit length of the "J" field is implied by the "S" and "L" fields in the XEC, NZT, and XMIT instructions, based on the following conditions:

- When the Source (S) field specifies an internal register, the literal value of the "J" field is an 8-bit binary number.
- When the Source (S) field specifies a variable I/O data field, the literal value of the "J" field is a 5-bit binary number.

A Field. The 13-bit "A" field is an address field which allows the 8X305 to directly branch to any of the 8192 locations in Program Storage memory.

Formation of Instruction Address

The Address Register and Program Counter are used to generate addresses for accessing an instruction from program storage. The instruction address is formed in one of the following ways:

- For all except the JMP, XEC, and a "satisfied" NZT instruction, the Program Counter is incremented by one and placed in the Address Register.
- For the JMP instruction, the 13-bit "A" field contained in the JMP instruction word replaces the contents of both the Address Register and the Program Counter.
- For the XEC instruction, the Address Register is loaded with bits from the Program Counter modified as follows:

  XEC using $\overline{IV}$ Bus Data — low-order 5 bits of ALU output replaces counterpart bits in Address Register

  XEC using Data from Internal Register — low-order 8 bits of ALU output replaces counterpart bits in Address Register

  The Program Counter is not modified for either of the above conditions.
- For a "satisfied" NZT instruction, the low-order 5 bits (NZT source is $\overline{IV}$ bus data) or low-order 8 bits (NZT source is an internal register) of both the Address Register and Program Counter are loaded with the literal value specified by the "J" field of instruction word.

Data Addressing

The source and/or destination addresses of the data to be operated upon are specified as part of the instruction word. As shown earlier, source/destination addresses are specified using a 5-bit code (00_8-37_8). When the most significant octal digit is a "0" or "1", the source and/or destination address is an internal register; if the most significant digit is a 2 or 3, an $\overline{IV}$ bus operation is indicated — 2 specifying a Left-Bank ($\overline{LB}$) operation and 3 specifying a Right-Bank ($\overline{RB}$) operation. The least significant octal digit (0 through 7) indicates either a specific internal register address or positioning information for the least significant bit when specifying $\overline{IV}$ bus data. Referring to Table 1, AUXiliary register R0 (00_8) is the implied source of the second argument for the ADD, AND, and XOR operations. IVL register R7 and IVR register R17 (destination addresses 07_8 and 17_8, respectively) provide a means of routing enabling address information to I/O peripherals. With IVL or IVR specified as the destination address, data is placed on the $\overline{\text{IV}}$ bus during the output phase of the instruction cycle; simultaneously, a Select Command (SC) is generated to inform all I/O devices that information on the $\overline{\text{IV}}$ bus is to be considered as an I/O address. Since the contents of IVL and IVR are preserved, either register may later be accessed as a source of data.

Control outputs $\overline{\text{LB}}$ and $\overline{\text{RB}}$ are used to partition I/O bus devices into two fields of 256 addresses. With $\overline{\text{LB}}$ in the active-low state and a source address of 20_8-27_8, the left bank of I/O devices are enabled during the input phase of the instruction cycle. With $\overline{\text{RB}}$ in the active-low state and a source address of 30_8-37_8, the right bank of devices are enabled. During the output phase, $\overline{\text{LB}}$ is low if the destination address is 07_8 or 20_8-27_8, whereas $\overline{\text{RB}}$ is low if the destination address is 17_8 or 30_8-37_8. Each address field ($\overline{\text{LB}}$ and $\overline{\text{RB}}$) can have a different I/O device selected, that is, data can be transferred from a device in one bank to a device in the other in one instruction cycle.

DESIGN PARAMETERS

Hardware design of an 8X305-based system largely consists of the following operations:

- Selecting and interfacing a Program Storage device — ROM, PROM, etc.
- Selecting and interfacing input/output devices — RAM, Ports, and other 8-bit addressable I/O devices.
- Choosing and implementing System Clock — Capacitor-Controlled, Crystal-Controlled, or Externally-Driven.
- Selection of an off-chip series-pass transistor.

All information required for easy implementation of these design requirements is provided under the folowing captions:

- DC Characteristics
- AC Characteristics
- Timing Considerations
- Clock Considerations
- $\overline{\text{HALT}}/\overline{\text{RESET}}$ Logic
- Voltage Regulator

DC CHARACTERISTICS (Commercial Part) $4.75V \leq V_{CC} \leq 5.25V$, $0°C \leq T_A \leq 70°C$

ABSOLUTE MAXIMUM RATINGS

PARAMETER	DESCRIPTION	RATING	UNIT	PARAMETER	DESCRIPTION	RATING	UNIT
V_{CC}	Supply voltage	+7.0	V	Others	Logic input voltage	5.5	V
X1, X2	Crystal input voltage	2.0	V	T_{STG}	Storage temperature	−65 to +150	°C

PARAMETER		TEST CONDITIONS	LIMITS			UNIT	COMMENTS
			Min	Typ	Max		
V_{CC}	Supply voltage		4.75	5.0	5.25	V	
V_{IH}	High level input voltage		0.6 2.0		2.0	V	X1 and X2 All other pins
V_{IL}	Low level input voltage				0.4 0.8	V	X1 and X2 All other pins
V_{OH}	High level output voltage	V_{CC} = min; I_{OH} = −3mA	2.4			V	
V_{OL}	Low level output voltage	V_{CC} = min; I_{OL} = 6mA V_{CC} = min, I_{OL} = 16mA			0.55 0.55	V	A_0 through A_{12} All other outputs
V_{CR}	Regulator voltage	V_{CC} = 5V		3.1 2.9		V	T_A = 0°C T_A = 70°C
V_{IC}	Input clamp voltage	V_{CC} = min, I_{IN} = −10mA			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes
I_{IH}	High level input current	V_{CC} = max V_{IH} = 0.6V V_{IH} = 4.5V			30 50	mA µA	X1 and X2 All other pins
I_{IL}	Low-level input current	V_{CC} = max; V_{IL} = 0.4V			−3 −0.2 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$-$\overline{IV7}$ I0-I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS}	Short circuit output current	V_{CC} = max; (Note: At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC}	Supply current	V_{CC} = max			180 195	mA	T_A = 70°C T_A = 0°C
I_{REG}	Regulator control	V_{CC} = 5.0V			−25	mA	Max available base drive for series-pass transistor
I_{CR}	Regulator current	V_{CC} = max			200 230	mA	T_A = 70°C T_A = 0°C

Notes
1 Operating temperature ranges are guaranteed after thermal equilibrium has been reached
2 All voltages measured with respect to ground terminal

AC CHARACTERISTICS (Commercial Part) CONDITIONS: $4.75V \leq V_{CC} \leq 5.25V$; $0°C \leq T_A \leq 70°C$

LOADING: (See test circuits)

PARAMETER (Note 1)		LIMITS (INSTRUCTION CYCLE TIME = 200ns)			LIMITS (INSTRUCTION CYCLE TIME > 200ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{PC}	Processor cycle time	200			200			ns	
T_{CP}	X1 clock period	100			100			ns	
T_{CH}	X1 clock high time	50			50			ns	
T_{CL}	X1 clock low time	50			50			ns	
T_{MCH}	MCLK high delay	20		35	20		35	ns	
T_{MCL}	MCLK low delay	15		30	15		30	ns	
T_W	MCLK pulse width	40		60	T_{40} − 10		T_{40} + 10	ns	Note 2
T_{AS}	X1 falling edge to address stable			55			55	ns	Note 7

AC CHARACTERISTICS (Commercial Part) CONDITIONS: 4.75V ≤ V_{CC} ≤ 5.25V; 0°C ≤ T_A ≤ 70°C
(Continued) LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 200ns)			LIMITS (INSTRUCTION CYCLE TIME > 200ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{MAS}	MCLK falling edge to address stable			140			T_{1Q} + T_{2Q} + 40	ns	Notes 2, 3, & 7
T_{IA}	Instruction to address			140			T_{2Q} + 90	ns	Notes 2, 3, & 8
T_{IVA}	Input data to address			85			85	ns	Notes 3 & 9
T_{IS}	Instruction set up time (X1 rising edge)	−5			−5			ns	Note 10
T_{MIS}	MCLK falling edge to instruction stable			30			T_{1Q} − 20	ns	Notes 2, 4, & 10
T_{IH}	Instruction hold time (X1 rising edge)	25			25			ns	Note 11
T_{MIH}	Instruction hold time (MCLK falling edge)	55			T_{1Q} + 5			ns	Notes 2 & 11
T_{WH}	X1 falling edge to SC/WC rising edge	25		45	25		45	ns	
T_{MWH}	MCLK falling edge to SC/WC rising edge	105		125	T_{1Q} + T_{2Q} + 5		T_{1Q} + T_{2Q} + 25	ns	Note 2
T_{WL}	X1 falling edge to SC/WC falling edge	25		35	25		35	ns	
T_{MWL}	MCLK falling edge to SC/WC falling edge	5		15	5		15	ns	
T_{IBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	30		45	30		45	ns	
T_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	10		25	10		25	ns	
T_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)			25			25	ns	
T_{OBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	35		60	35		60	ns	
T_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	115		145	T_{1Q} + T_{2Q} + 15		T_{1Q} + T_{2Q} + 45	ns	Note 2
T_{IDS}	Input data set up time (X1 falling edge)	25			25			ns	
T_{MIDS}	MCLK falling edge to input data stable			55			T_{1Q} + T_{2Q} − 45	ns	Notes 2 & 5
T_{IDH}	Input data hold time (X1 falling edge)	35			35			ns	
T_{MIDH}	Input data hold time (MCLK falling edge)	115			T_{1Q} + T_{2Q} + 15			ns	Note 2
T_{ODH}	Output data hold tim (X1 falling edge)	30			30			ns	
T_{MODH}	Output data hold time (MCLK falling edge)	11			11			ns	
T_{ODS}	Output data stable (X1 falling edge)	50		70	50		70	ns	Notes 12 & 15
T_{MODS}	Output data stable (MCLK falling edge)	130		150	T_{1Q} + T_{2Q} + 30		T_{1Q} + T_{2Q} + 50	ns	Notes 2, 12, & 15

AC CHARACTERISTICS (Commercial Part) (Continued)

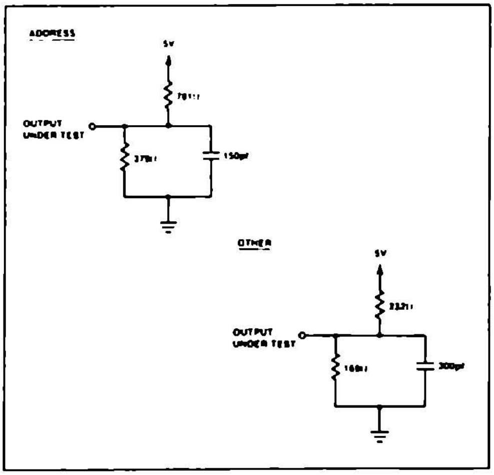
CONDITIONS: 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 200ns)			LIMITS (INSTRUCTION CYCLE TIME > 200ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{CDO}	Output driver turn-on time (X1 falling edge)	45		65	45		65	ns	Note 14
T_{MODO}	Output driver turn-on time (MCLK falling edge)	125		145	T_{1Q} + T_{2Q} − 25		T_{1Q} + T_{2Q} + 45	ns	Note 14
T_{DI}	Output driver turn-on time (SC/WC rising edge)	20			20			ns	Note 16
T_{DD}	Input data to output data	85		105	85		105	ns	Notes 13 & 15
T_{HS}	$\overline{HALT}$ set up time (X1 rising edge)	0			0			ns	
T_{MHS}	MCLK falling edge to $\overline{HALT}$ falling edge			30			T_{1Q} − 20	ns	Notes 2 & 6
T_{HH}	$\overline{HALT}$ hold time (X1 rising edge)	35			35			ns	
T_{MHH}	$\overline{HALT}$ hold time (MCLK falling edge)	65			T_{1Q} + 15			ns	Note 2
T_{ACC}	Program storage access time			60				ns	
T_{IO}	I/O port output enable time ($\overline{LB}/\overline{RB}$ to valid IV data input)			30				ns	

Notes

1. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts; all timing parameters are measured at this voltage level.
2. Respectively, T_{1Q}, T_{2Q}, T_{3Q} and T_{4Q} represent time intervals for the first, second, third and fourth quarter cycles.
3. Capacitive loading for the address bus is 150 picofarads.
4. Same as TIS but referenced to falling edge of MCLK.
5. Same as TIDS but referenced to falling edge of MCLK.
6. Same as THS but referenced to falling edge of MCLK.
7. TAS is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set up time. The TAS parameter then represents the earliest time that the address bus is valid.
8. TIA is obtained by forcing a valid instruction input to occur earlier than the minimum set up time.
9. TIVA is obtained by forcing a valid I/O bus input to just meet the minimum set up time.
10. TMIS represents the set up time required by internal latches of the 8X305. In system applications, the instruction input may have to be valid before the worst case set up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set up time (TIDS and TMIDS).
11. TIH represents the hold time required by internal latches of the 8X305. To generate proper $\overline{LB}/\overline{RB}$ signals, the instruction must be held valid until the address bus changes.
12. TODS is obtained by forcing a valid I/O bus input to occur earlier than the I/O bus input set up time (TIDS). This timing parameter represents the earliest time that the I/O output data can be valid.
13. TOD is obtained by forcing a valid I/O bus input to just meet the minimum I/O bus input set up time. This timing parameter represents the latest time that the I/O output data can be valid.
14. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the 8X305 will turn on.
15. For TIDS ≥ 25ns, TODS or TMODS should be used to determine when the output data is stable.
16. This parameter represents the latest time that the output drivers of the input device should be turned off.

TEST CIRCUITS



DC CHARACTERISTICS (Military Part) $4.5V < V_{CC} < 5.5V$, $-55°C < T_C < +125°C$

ABSOLUTE MAXIMUM RATINGS

PARAMETER	DESCRIPTION	RATING	UNIT	PARAMETER	DESCRIPTION	RATING	UNIT
V_{CC} X1, X2	Supply voltage Crystal input voltage	+7.0 2.0	V V	Others T_{STG}	Logic input voltage Storage temperature	5.5 −65 to +150	V °C

PARAMETER	TEST CONDITIONS	LIMITS Min	Typ	Max	UNIT	COMMENTS
V_{CC} Supply voltage		4.5	5.0	5.5	V	
V_{IH} High level input voltage		0.6 2.0		2.0	V	X1 and X2 All other pins
V_{IL} Low level input voltage				0.4 0.8	V	X1 and X2 All other pins
V_{OH} High level output voltage	V_{CC} = min; I_{OH} = −3mA	2.4			V	
V_{OL} Low level output voltage	V_{CC} = min; I_{OL} = 6mA V_{CC} = min; I_{OL} = 16mA			0.55 0.55	V	A_0 through A_{12} All other outputs
V_{CR} Regulator voltage	V_{CC} = 5V		3.5 3.1 2.6		V	T_C = −55°C T_C = 0°C T_C = 125°C
V_{IC} Input clamp voltage	V_{CC} = min; I_{IN} = −10mA			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes
I_{IH} High level input current	V_{CC} = max V_{IH} = 0.6V V_{IH} = 4.5V			3.0 50	mA μA	X1 and X2 All other pins
I_{IL} Low-level input current	V_{CC} = max; V_{IL} = 0.4V			−3 −0.3 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$-$\overline{IV7}$ I0-I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS} Short circuit output current	V_{CC} = max; (Note: At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC} Supply current	V_{CC} = max			175 205	mA	T_C = 125°C T_C = −55°C
I_{REG} Regulator control	V_{CC} = 5.0V			−25	mA	Max available base drive for series-pass transistor
I_{CR} Regulator current	V_{CC} = max			180 260	mA	T_C = 125°C T_C = −55°C

Notes.

1 Operating temperature ranges are guaranteed after thermal equilibrium has been reached

2 All voltages measured with respect to ground terminal

AC CHARACTERISTICS (Military Part) CONDITIONS: $4.5V < V_{CC} < 5.5V$; $-55°C < T_C < 125°C$

LOADING: (See test circuits)

PARAMETER (Note 1)		LIMITS (INSTRUCTION CYCLE TIME = 250ns) Min	Typ	Max	LIMITS (INSTRUCTION CYCLE TIME > 250ns) Min	Typ	Max	UNITS	COMMENTS
T_{PC}	Processor cycle time	250			250			ns	
T_{CP}	X1 clock period	125			125			ns	
T_{CH}	X1 clock high time	62			62			ns	
T_{CL}	X1 clock low time	62			62			ns	
T_{MCH}	MCLK high delay	15		35	15		35	ns	
T_{MCL}	MCLK low delay	15		30	15		30	ns	
T_W	MCLK pulse width	52		72	T_{40} − 10		T_{40} + 10	ns	Note 2
T_{AS}	X1 falling edge to address stable			60			60	ns	Note 7

AC CHARACTERISTICS (Military Part) CONDITIONS: 4.5V < V_{CC} < 5.5V, −55°C < T_C < 125°C
(Continued) LOADING: (See test circuits)

PARAMETER (Note 1)		LIMITS (INSTRUCTION CYCLE TIME = 250ns)			LIMITS (INSTRUCTION CYCLE TIME > 250ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{MAS}	MCLK falling edge to address stable			160			T_{1O} + T_{2O} + 35	ns	Notes 2, 3, & 7
T_{IA}	Instruction to address			160			T_{2O} + 98	ns	Notes 2, 3, & 8
T_{IVA}	Input data to address			90			90	ns	Notes 3 & 9
T_{IS}	Instruction set up time (X1 rising edge)	0			0			ns	Note 10
T_{MIS}	MCLK falling edge to instruction stable			40			T_{1O} − 22	ns	Notes 2, 4, & 10
T_{IH}	Instruction hold time (X1 rising edge)	30			30			ns	Note 11
T_{MIH}	Instruction hold time (MCLK falling edge)	70			T_{1O} + 8			ns	Notes 2 & 11
T_{WH}	X1 falling edge to SC/WC rising edge	25		50	25		50	ns	
T_{MWH}	MCLK falling edge to SC/WC rising edge	127		154	T_{1O} + T_{2O} + 2		T_{1O} + T_{2O} + 29	ns	Note 2
T_{WL}	X1 falling edge to SC/WC falling edge	15		35	15		35	ns	
T_{MWL}	MCLK falling edge to SC/WC falling edge	5		25	5		25	ns	
T_{IBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	25		55	25		55	ns	
T_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	10		35	10		35	ns	
T_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)			30			30	ns	
T_{OBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	35		65	35		65	ns	
T_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	140		170	T_{1O} + T_{2O} + 15		T_{1O} + T_{2O} + 45	ns	Note 2
T_{IDS}	Input data set up time (X1 falling edge)	30			30			ns	
T_{MIDS}	MCLK falling edge to input data stable			75			T_{1O} + T_{2O} − 50	ns	Notes 2 & 5
T_{IDH}	Input data hold time (X1 falling edge)	35			35			ns	
T_{MIDH}	Input data hold time (MCLK falling edge)	140			T_{1O} + T_{2O} + 15			ns	Note 2
T_{ODH}	Output data hold time (X1 falling edge)	25			25			ns	
T_{MODH}	Output data hold time (MCLK falling edge)	11			11			ns	
T_{ODS}	Output data stable (X1 falling edge)	50		80	50		80	ns	Notes 12 & 15
T_{MODS}	Output data stable (MCLK falling edge)	150		180	T_{1O} + T_{2O} + 25		T_{1O} + T_{2O} + 55	ns	Notes 2, 12, & 15

AC CHARACTERISTICS (Military Part) (Continued)

CONDITIONS: 4.5V < V_{CC} < 5.5V; – 55°C < T_C < 125°C
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 250ns)			LIMITS (INSTRUCTION CYCLE TIME > 250ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{ODO}	Output driver turn-on time (X1 falling edge)	45		70	45		70	ns	Note 14
T_{MODO}	Output driver turn-on time (MCLK falling edge)	145		175	T_{1Q} + T_{2Q} + 20		T_{1Q} + T_{2Q} + 50	ns	Note 14
T_{DI}	Output driver turn-on time (SC/WC rising edge)	20			20			ns	Note 16
T_{DD}	Input data to output data	80		115	80		115	ns	Notes 13 & 15
T_{HS}	$\overline{\text{HALT}}$ set up time (X1 rising edge)	0			0			ns	
T_{MHS}	MCLK falling edge to $\overline{\text{HALT}}$ falling edge			40			T_{1Q} – 22	ns	Notes 2 & 6
T_{HH}	$\overline{\text{HALT}}$ hold time (X1 rising edge)	35			35			ns	
T_{MHH}	$\overline{\text{HALT}}$ hold time (MCLK falling edge)	80			T_{1Q} + 18			ns	Note 2
T_{ACC}	Program storage access time			90				ns	
T_{IO}	I/O port output enable time ($\overline{\text{LB}}$/$\overline{\text{RB}}$ to valid $\overline{\text{IV}}$ data input)			40				ns	

Notes

1 X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts, all timing parameters are measured at this voltage level

2 Respectively, T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles

3 Capacitive loading for the address bus is 150 picofarads

4 Same as TIS but referenced to falling edge of MCLK

5 Same as TIDS but referenced to falling edge of MCLK

6 Same as THS but referenced to falling edge of MCLK

7 TAS is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set up time. the TAS parameter then represents the earliest time that the address bus is valid

8 TIA is obtained by forcing a valid instruction input to occur earlier than the minimum set up time

9 TIVA is obtained by forcing a valid I/O bus input to just meet the minimum set up time

10 TMIS represents the set up time required by internal latches of the 8X305. In system applications, the instruction input may have to be valid before the worst-case set up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set up time (TIDS and TMIDS)

11 TIH represents the hold time required by internal latches of the 8X305. To generate proper $\overline{\text{LB}}$/$\overline{\text{RB}}$ signals, the instruction must be held valid until the address bus changes

12 TODS is obtained by forcing a valid I/O bus input to occur earlier than the I/O bus input set up time (TIDS), this timing parameter represents the earliest time that the I/O output data can be valid

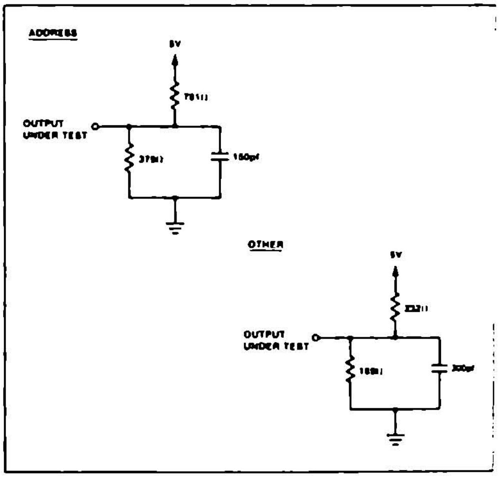
13 TDD is obtained by forcing a valid I/O bus input to just meet the minimum I/O bus input set up time. this timing parameter represents the latest time that the I/O output data can be valid

14 The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the 8X305 will turn on

15 For TIDS > 30ns, TODS or TMODS should be used to determine when the output data is stable

16 This parameter represents the latest time that the output drivers of the input device should be turned off

TEST CIRCUITS



TIMING CONSIDERATIONS (Commercial Part)

As shown in the AC CHARACTERISTICS table for the commercial part, the minimum instruction cycle time is 200 nanoseconds; whereas, the maximum is determined by the on-chip oscillator frequency and can be any value the user chooses. With an instruction cycle time of 200 nanoseconds, the part can be characterized in terms of absolute values; these are shown in the first "LIMITS" column of the table. When the instruction cycle time is greater than 200 nanoseconds, certain parameters are cycle-time dependent; thus, these parameters are specified in terms of the four quarter cycles (T_{1O}, T_{2O}, T_{3O}, and T_{4O}) that make up one instruction cycle — see 8X305 TIMING DIAGRAM. As the time interval for each instruction cycle increases (becomes greater than 200 nanoseconds), the delay for all parameters that are cycle-time dependent is likewise increased. In some cases, these delays have a significant impact on timing relationships and other areas of systems design; subsequent paragraphs describe these timing parameters and reliable methods of calculation.

Timing parameters for the 8X305 are normally measured with reference to X1 or MCLK; those referenced to MCLK are prefaced with an "M" in the mnemonic — TMAS, TMIH, and so on. To determine the timing relationship between a particular signal, say "A" and MCLK, the user should, at all times, use the value specified in the table — DO NOT calculate the value by adding or subtracting two or more parameters that are referenced to X1. When deriving timing relationships between two signals (A to B, etc.) by adding or subtracting the parameter values, the user must consistently use the same parameter reference — MCLK or X1.

System determinants for the instruction cycle time are:

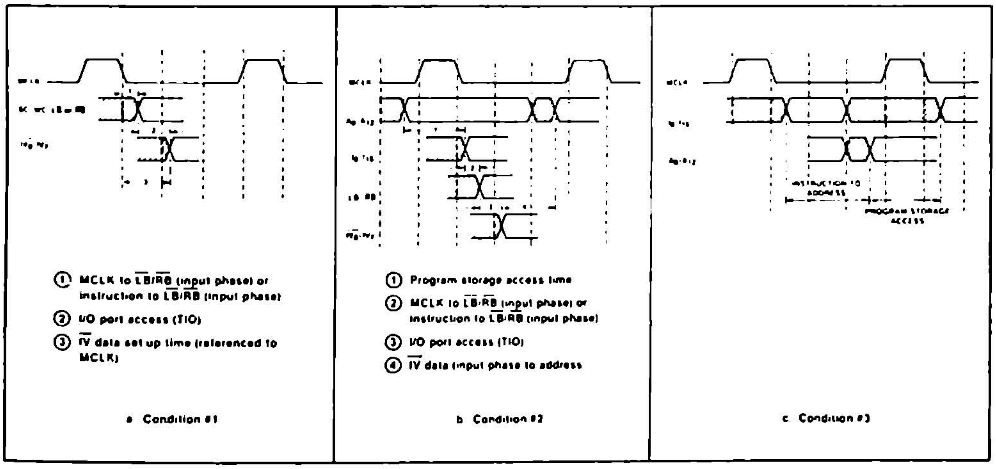
- Propagation delays within the 8X305
- Access time of Program Storage
- Enable time of the I/O port

Normally, the instruction cycle time is constrained by one or more of the following conditions.

Condition 1 — Instruction or MCLK to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase) plus I/O port access time (TIO) ≤ $\overline{\text{IV}}$ data set up time (Figure 5a).

Condition 2 — Program storage access time (TACC) plus instruction to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase) plus I/O port access time (TIO) plus $\overline{\text{IV}}$ data (input phase) to address ≤ instruction cycle time (Figure 5b).

Condition 3 — Program storage access time plus instruction to address ≤ instruction cycle time (Figure 5c).



① MCLK to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase) or instruction to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase)
② I/O port access (TIO)
③ $\overline{\text{IV}}$ data set up time (referenced to MCLK)

a. Condition #1

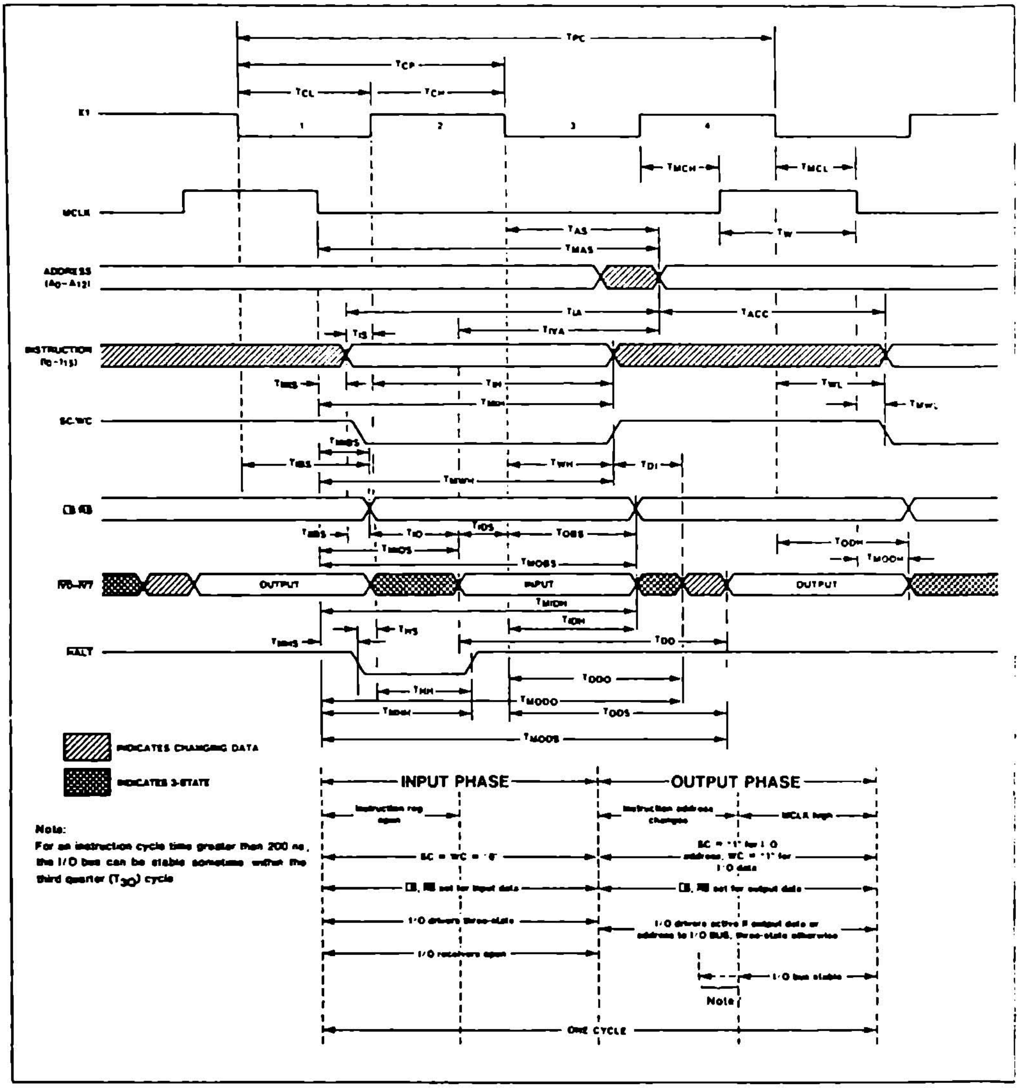
① Program storage access time
② MCLK to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase) or instruction to $\overline{\text{LB}}$/$\overline{\text{RB}}$ (input phase)
③ I/O port access (TIO)
④ $\overline{\text{IV}}$ data (input phase) to address

b. Condition #2

c. Condition #3

Figure 5. Constraints of 8X305 Instruction Cycle Time

8X305 TIMING DIAGRAM


X1
MCLK
ADDRESS (A0–A12)
INSTRUCTION (I0–I15)
SC/WC
LB, RB
IV0–IV7
HALT
TPC
TCP
TCL
TCH
TMCH
TMCL
TAS
TMAS
TW
TIA
TACC
TIS
TIVA
TIH
TWL
TMWL
TWH
TDI
TIO
TIOS
TOBS
TOOH
TMOOH
OUTPUT
INPUT
OUTPUT
TIOH
THS
TDO
THH
TMODO
TODO
TODS
TMODS
INDICATES CHANGING DATA
INDICATES 3-STATE
Note:
For an instruction cycle time greater than 200 ns, the I/O bus can be stable sometime within the third quarter (T3Q) cycle
INPUT PHASE
OUTPUT PHASE
Instruction reg open
Instruction address changes
MCLK high
SC = WC = "0"
SC = "1" for I/O address, WC = "1" for I/O data
LB, RB set for input data
LB, RB set for output data
I/O drivers three-state
I/O drivers active if output data or address to I/O BUS, three-state otherwise
I/O receivers open
I/O bus stable
Note
ONE CYCLE

From condition #1 and with an instruction cycle time of 200ns, the I/O port access time (TIO) can be calculated as follows:

TMIBS + TIO ≤ TMIDS
transposing, TIO ≤ TMIDS = TMIBS
substituting, TIO ≤ 55ns - 25ns
result, TIO ≤ 30ns

Using 30ns for TIO, the constraint imposed by condition #1 can also be used to calculate the minimum cycle time:

TMIBS + TIO ≤ TMIDS
thus, 25ns + 30ns ≤ $T_{1Q} + T_{2Q} - 45$
25ns + 30ns ≤ 1/2 cycle - 45 therefore,

the worst-case instruction cycle time is 200ns. With subject parameters referenced to X1, the same calculations are valid:

TIBS + TIO + TIDS ≤ 1/2 cycle
thus, 45ns + 30ns + 25ns ≤ 1/2 cycle therefore,

the worst-case instruction cycle time is again 200ns. From condition #2 and with an instruction cycle time of 200ns, the program storage access time can be calculated:

TACC + TIIBS + TIO + TIVA ≤ 200ns
transposing, TACC ≤ 200ns - TIIBS - TIO - TIVA
substituting, TACC ≤ 200ns - 25ns - 30ns - 85ns
thus, TACC ≤ 60ns hence, for an instruction

cycle time of 200ns, a program storage access time of 60ns is implied. The constraint imposed by condition #3 can be used to verify the maximum program storage access time:

TIA + TACC ≤ Instruction Cycle
thus, TACC ≤ 200ns - 140ns
and, TACC ≤ 60ns, confirming that a program

storage access time of 60ns is satisfactory.

For an instruction cycle time of 200ns and a program storage access time of 60ns (Condition #2/Figure 5b), the instruction should be valid at the falling edge of MCLK. This relationship can be derived by the following equation:

200ns - TMAS - TACC
= 200ns - 140ns - 60ns
= 0ns

It is important to note that, during the input phase, the beginning of a valid $\overline{LB}/\overline{RB}$ signal is determined by either the instruction to $\overline{LB}/\overline{RB}$ delay (TIIBS) or the delay from the falling edge of MCLK to $\overline{LB}/\overline{RB}$ (TMIBS). Assuming the instruction is valid at the falling edge of MCLK and adding the instruction-to-$\overline{LB}/\overline{RB}$ delay (TIIBS = 25ns), the $\overline{LB}/\overline{RB}$ signal will be valid 25ns after the falling edge of MCLK. With a fast program storage memory and with a valid instruction before the falling edge of MCLK — the $\overline{LB}/\overline{RB}$ signal will, due to the TMIBS delay, still be valid 25ns after the falling edge of MCLK. Using a worst-case instruction cycle time of 200ns, the user cannot gain a speed advantage by selecting a memory with faster access time. Under the same conditions, a speed advantage cannot be obtained by using an I/O port with fast access time (TIO) because the address bus will be stable 55ns (TAS) after the beginning of the third quarter cycle — no matter how early the $\overline{IV}$ data input is valid.

Internal Timing and Timing Relationships

All timing and timing-control signals of the 8X305 are generated by the oscillator and sequencer shown in Figure 6. Outputs from the sequencer direct and control all of the timing parameters specified in the TIMING DIAGRAM. Observe that each input quarter cycle bears a fixed relationship to X1 via the propagation delay.

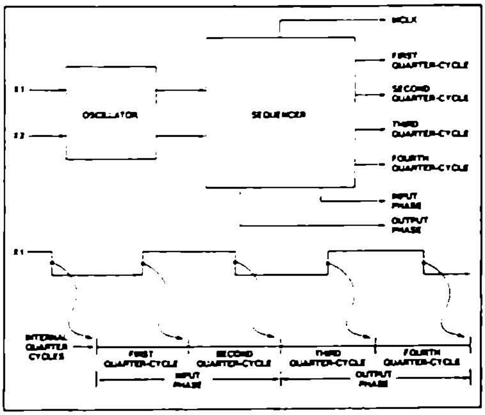


Figure 6. Timing and Timing Control Signals of the 8X305

General and interactive timing relationships pertaining to I/O signals of the 8X305 are shown in Figure 7. Example— in the input phase, the switching point of the $\overline{LB}/\overline{RB}$ signal is caused by the worst-case delay from the instruction to $\overline{LB}/\overline{RB}$ or from the beginning of the first internal quarter cycle to $\overline{LB}/\overline{RB}$; the two arrows pointing to the $\overline{LB}/\overline{RB}$ transition indicate this "either/or" dependency. This information coupled with tabular values and the TIMING DIAGRAM provides the user with the wherewithal to calculate any and all system timing parameters.

CLOCK CONSIDERATIONS

The on-chip oscillator and timing-generation circuits of the 8X305 can be controlled by any one of the following methods:

Capacitor — if timing is not critical
Crystal — if precise timing is required
External Drive — if application requires that the 8X305 be driven from a system clock

Capacitor Timing. A non-polarized ceramic or mica capacitor with a working voltage equal to or greater than 25 volts is recommended. The lead lengths of capacitor

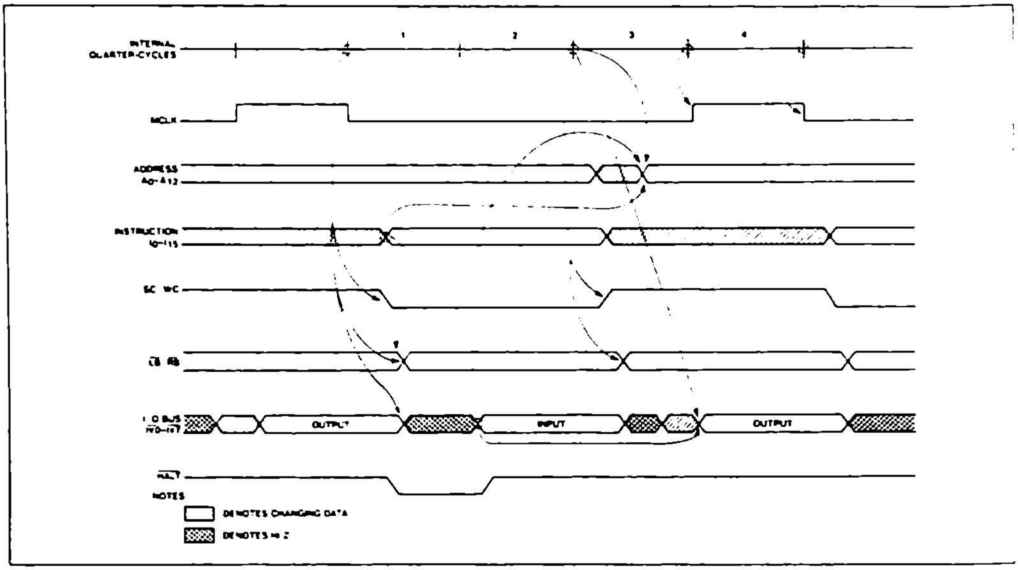

INTERNAL QUARTER-CYCLES
1
2
3
4
MCLK
ADDRESS A0-A12
INSTRUCTION I0-I15
SC WC
LB RB
I/O BUS IV0-IV7
OUTPUT
INPUT
OUTPUT
HALT
NOTES
DENOTES CHANGING DATA
DENOTES HI Z

Figure 7. Timing Relationships of 8X305 I/O Signals

should be approximately the same and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. For various capacitor (C_X) values, the cycle time can be approximated as:

C_X (In pF)	APPROXIMATE CYCLE TIME
100	300ns
200	500ns
500	1.1μs
1000	2.0μs

Crystal Timing. When a crystal is used, the on-chip oscillator operates at the resonant frequency (f_0) of the crystal; the series-resonant quartz crystal connects to the 8X305 via pins 10 (X1) and 11 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

Type — Fundamental mode, series resonant
Impedance at Fundamental — 35 ohms maximum
Impedance at Harmonics and Spurs — 50 ohms minimum

The resonant frequency (f_0) of the crystal is related to the desired cycle time (T) by the equation: $f_0 = 2/T$; thus, for a cycle time of 200 nanoseconds, $f_0 = 10MHz$. Other cycle times versus crystal frequencies are as follows:

CYCLE TIME (In ns)	CRYSTAL FREQUENCY (In MHz)
200	10.0
250	8.0
400	5.0
500	4.0
750	2.6
1000	2.0

Note:
Frequencies below 2MHz may not cause oscillations and are not recommended

Using an External Clock. The 8X305 can be synchronized with an external clock by simply connecting appropriate drive circuits to the X1/X2 inputs. Figure 8 shows how the on-chip oscillator can be driven from the complementary outputs of a pulse generator. In applications where the microcontroller must be driven from a master clock, the X1/X2 lines can be interfaced to TTL logic as shown in Figure 9.

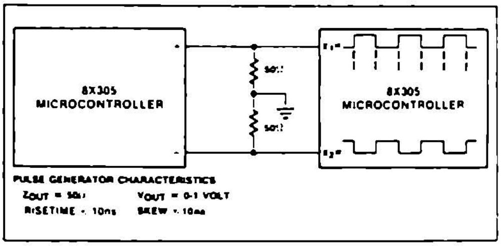

8X305
MICROCONTROLLER
50Ω
50Ω
X1
X2
8X305
MICROCONTROLLER
PULSE GENERATOR CHARACTERISTICS
ZOUT = 50Ω
VOUT = 0-1 VOLT
RISETIME < 10ns
SKEW < 10ns

Figure 8. Clocking with a Pulse Generator

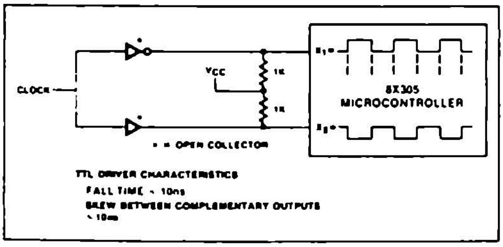

CLOCK
VCC
1K
1K
X1
X2
8X305
MICROCONTROLLER
* = OPEN COLLECTOR
TTL DRIVER CHARACTERISTICS
FALL TIME < 10ns
SKEW BETWEEN COMPLEMENTARY OUTPUTS
< 10ns

Figure 9. Clocking with TTL

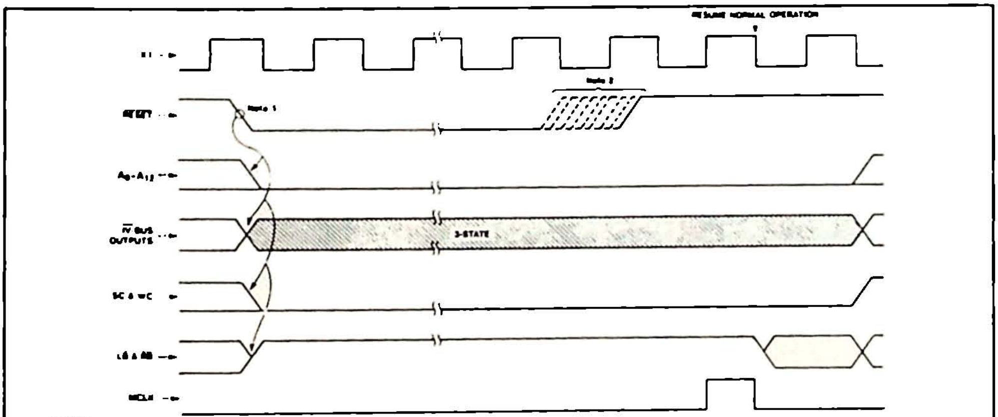
RESET Logic

RESET (pin 43) can be driven from a high (inactive) state to a low (active) state at any time with respect to the system clock, that is, the reset function is asynchronous. To ensure proper operation, RESET must be held low (active) for one full instruction time. When the line is driven from a high state to an active-low state, several events occur — the precise instant of occurrence is basically a function of the propagation delay for that particular event. As shown in the RESET TIMING DIAGRAM, these events are:

- The Program Counter and Address Register are set to address zero and remain in that state as long as the RESET line is low. Other than PC and AR, RESET does not affect other internal registers.
- The input/output (IV) bus goes three-state and remains in that condition as long as the RESET line is low.
- The Select Command and Write Command signals are driven low and remain low as long as the RESET line is low.
- The Left Bank/Right Bank (LB/RB) signals are forced high asynchronously for the period in which the RESET line is low.

During the time RESET is active-low, MCLK is inhibited, moreover, if the RESET line is driven low during the last two quarter cycles, MCLK may be shortened for that particular machine cycle. When RESET line is driven high (inactive)—one quarter to one full instruction cycle later, MCLK appears just before normal operation is resumed. The RESET/MCLK relationship is clearly shown by "B" in the timing diagram. As long as the RESET line is active-low, the HALT signal (described next) is not sampled by internal logic of the 8X305.

RESET TIMING DIAGRAM

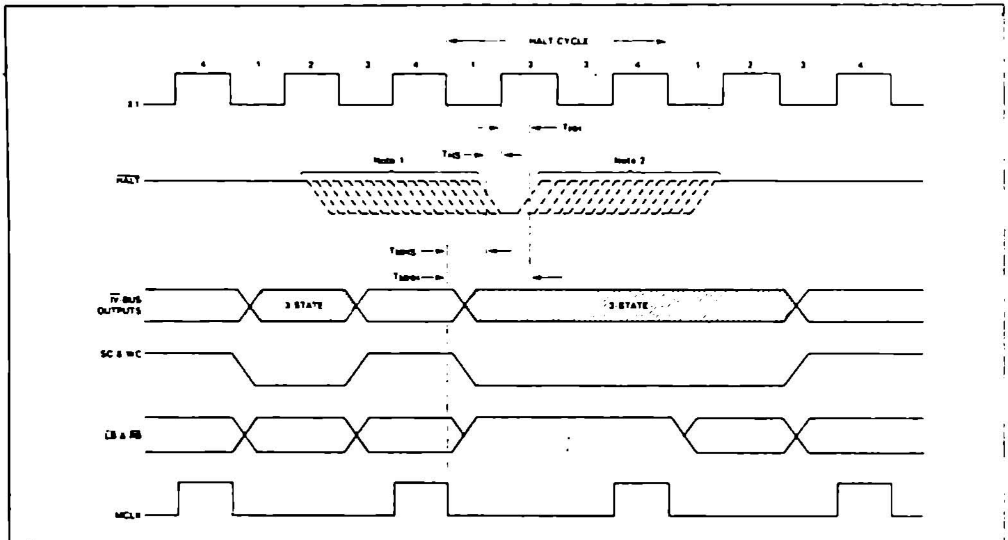

RESUME NORMAL OPERATION
X1
Note 2
RESET
Note 1
A0-A12
IV BUS OUTPUTS
3-STATE
SC & WC
LB & RB
MCLK

Notes:
1 A High to Low transition of the RESET signal will force the Address Bus to an all zero configuration
2 The RESET signal can switch from Low to High at any point within this time interval and, in all cases MCLK will occur at least one-quarter cycle time later as shown

$\overline{\text{HALT}}$ Logic

The $\overline{\text{HALT}}$ signal is sampled via internal chip logic at the end of the first internal quarter of each instruction cycle. If, when sampled, the $\overline{\text{HALT}}$ signal is active-low, a halt is immediately executed and the current instruction cycle is terminated; however, the halt cycle does not inhibit MCLK nor does it affect any internal registers of the 8X305. As long as the $\overline{\text{HALT}}$ line is active-low, the SC and WC lines are low (inactive), the Left Bank ($\overline{\text{LB}}$)/Right Bank (RB) signals are high (inactive), and the $\overline{\text{IV}}$ bus remains in the three-state mode of operation. Normal operation resumes at the next cycle in which $\overline{\text{HALT}}$ is high when sampled — see $\overline{\text{HALT}}$ TIMING DIAGRAM.

HALT TIMING DIAGRAM



Notes:

1. The $\overline{\text{HALT}}$ signal can switch from High to Low at any time during this interval.
2. The $\overline{\text{HALT}}$ signal can switch from Low to High at any time during this interval.

Timing Descriptions

T_{HS}—set-up time from $\overline{\text{HALT}}$ to X1 (independent of instruction cycle time)

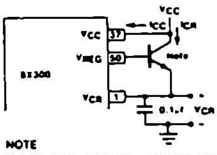
T_{HH}—hold time from X1 to $\overline{\text{HALT}}$ (independent of instruction cycle time)

T_{MHS}—set-up time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

T_{MHH}—hold time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

VOLTAGE REGULATOR

All internal logic of the 8X305 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in the accompanying diagram. To minimize lead inductance, the transistor should be as close as possible to the 8X305 package and the emitter should be ac-grounded via a 0.1 microfarad ceramic capacitor.



PARAMETER	CONDITIONS	LIMITS
h_{FE}	V_{CE} = 2V 100mA I_C 500mA	50
$V_{BE(ON)}$	V_{CE} = 5V I_C = 500mA	1V
V_{CESAT}	I_C = 500mA, I_B = 50mA	0.5V
BV_{CEO}		15V
f_T		30MHz

NOTE
Typical approved parts—2N5320, 2N5337

ORDERING INFORMATION

Commercial
N8X305N (Plastic)
N8X305I (Ceramic)

Military
S8X305I/883B
S8X305I/883C

JANUARY 1982

INTERRUPT CONTROL COPROCESSOR 8X310

FEATURES

- THREE MASKABLE, PRIORITIZED INTERRUPTS
- SUBROUTINE CONTROL CAPABILITY
- 4 LEVEL STACK FOR RETURN ADDRESSES
- DIRECTLY COMPATIBLE WITH THE 8X305 MICROCONTROLLER
- BIPOLAR ISL AND LOW-POWER SCHOTTKY TECHNOLOGY
- SINGLE +5 VOLT POWER SUPPLY
- 40-PIN DUAL IN-LINE PACKAGE

PRODUCT DESCRIPTION

The 8X310 Interrupt Control Coprocessor provides all of the circuitry required to enable the efficient use of the 8X300 MicroController Family in such interrupt driven environments as real time control systems. In addition, the address control capabilities of the device can be used to support subroutine handling.

Five instructions are supported by the Interrupt Control Coprocessor, all of which are treated as NOPs by the 8X305. It recognizes these by monitoring the 8X305 Instruction Bus at all times. Normal 8X305 operations are treated as NOPs by the coprocessor.

The 8X310 has three prioritized interrupt pins for interface to the user system. When one of these pins receives an interrupt signal, the device performs two actions:

- The address of the next sequential instruction to be executed is stored in a 4 level pushdown stack in the 8X310 to enable return to normal processing after the interrupt has been handled.
- A JUMP instruction is placed on the 8X305's Instruction Bus which forces transfer to an assigned memory location that corresponds to the interrupt pin that initiated the action. The assigned memory location, in turn, can contain the address of the interrupt handling software.

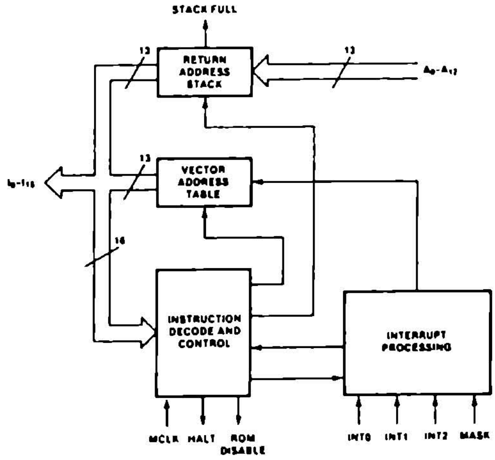
Since the device operates synchronously with the MicroController, the actions are completed in two processor cycles, or as little as 400 nanoseconds.

The 8X310 can also be used to efficiently handle subroutine calls and returns in an 8X305 based system. To facilitate this, an instruction is provided on the coprocessor that causes it to store the next instruction address in its stack witout an interrupt taking place.

Return to normal processing after an interrupt or to the calling program after subroutine execution is accomplished through a coprocessor instruction that forces a JUMP instruction to the next address in the stack onto the 8X305 Instruction Bus.

The 8X310 has additional instructions to set and clear interrupt masks and permit clearing of pending interrupts.

8X310 BLOCK DIAGRAM



ARCHITECTURAL OVERVIEW

The Signetics 8X320 Bus Interface Register Array (Figure 1) is a dual port RAM memory designed for use between a host processor and a peripheral processor. Specifically, the register array provides a convenient and economical interface between the 8X305 (or 8X300) Microcontroller (secondary port) and User's Host System (primary port). The host can be almost any bus-oriented device—another processor, a minicomputer or a mainframe computer. The host has 8-bit (byte) or 16-bit (word) access to the primary port, data can be read from or written into any memory location as determined by the primary port address and control lines. The secondary port (8X305 bus) consists of eight input/output lines and four bus control lines. To implement the secondary-port interface, an 8 bit memory location is addressed during one machine cycle and, during another cycle, data is read or written under control of the secondary (8X305) processor. Both primary and secondary ports feature three-state outputs and both ports are bidirectional.

Besides the convenience and economy of a two port memory, the array also provides simple handshake control via two 8-bit flag registers, logic to facilitate DMA transfers, and a write protect feature for the primary port in both byte and word modes of operation.

FEATURES

- **16-byte/2-port interface**
- **8- or 16-bit primary port (Host) interface (User selectable)**
- **8-bit secondary port interface**
- **Two 8-bit flag registers (handshake control)**
- **DMA or programmed I/O operation**
- **Two three-state bidirectional ports**
- **Secondary port is bus compatible with 8X305**
- **Single 5V supply**
- **40-pin package**

BLOCK DIAGRAM

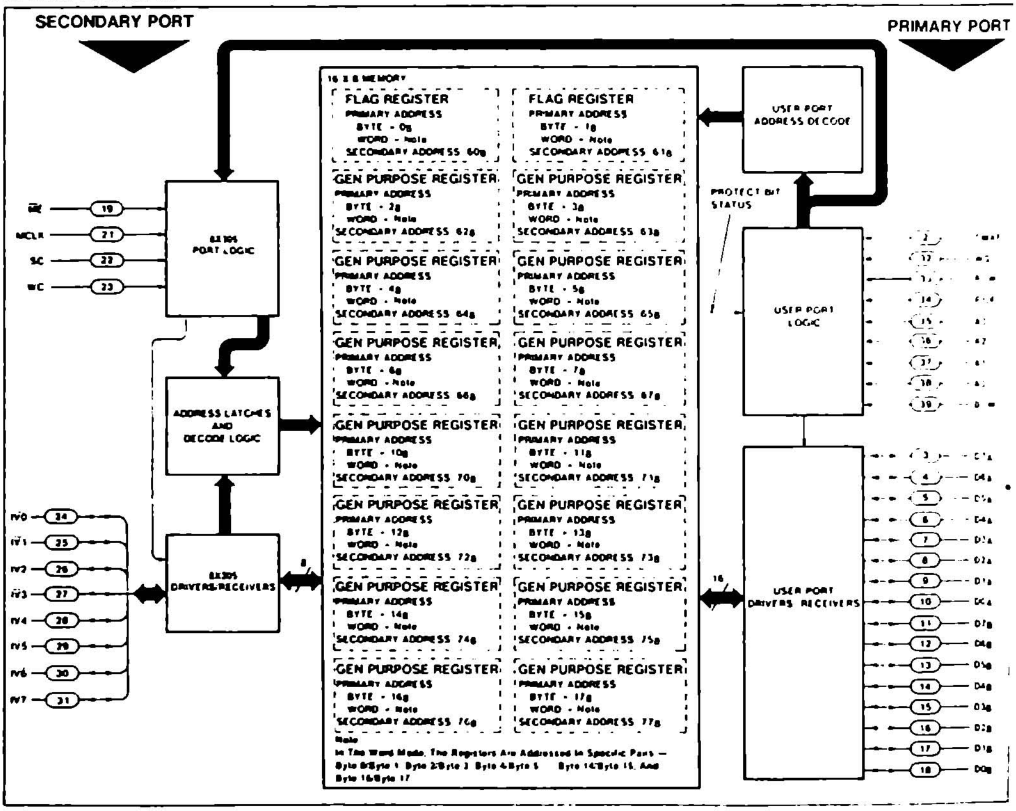
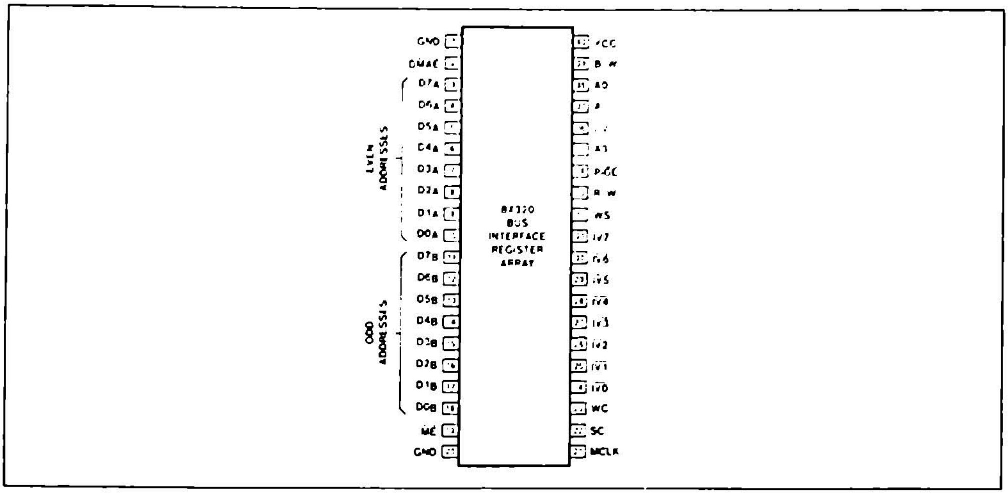

SECONDARY PORT
PRIMARY PORT
16 X 8 MEMORY
FLAG REGISTER
PRIMARY ADDRESS
BYTE - 0₈
WORD - Note
SECONDARY ADDRESS 60₈
FLAG REGISTER
PRIMARY ADDRESS
BYTE - 1₈
WORD - Note
SECONDARY ADDRESS 61₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 2₈
WORD - Note
SECONDARY ADDRESS 62₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 3₈
WORD - Note
SECONDARY ADDRESS 63₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 4₈
WORD - Note
SECONDARY ADDRESS 64₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 5₈
WORD - Note
SECONDARY ADDRESS 65₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 6₈
WORD - Note
SECONDARY ADDRESS 66₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 7₈
WORD - Note
SECONDARY ADDRESS 67₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 10₈
WORD - Note
SECONDARY ADDRESS 70₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 11₈
WORD - Note
SECONDARY ADDRESS 71₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 12₈
WORD - Note
SECONDARY ADDRESS 72₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 13₈
WORD - Note
SECONDARY ADDRESS 73₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 14₈
WORD - Note
SECONDARY ADDRESS 74₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 15₈
WORD - Note
SECONDARY ADDRESS 75₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 16₈
WORD - Note
SECONDARY ADDRESS 76₈
GEN PURPOSE REGISTER
PRIMARY ADDRESS
BYTE - 17₈
WORD - Note
SECONDARY ADDRESS 77₈
Note
In The Word Mode, The Registers Are Addressed In Specific Pairs — Byte 0/Byte 1, Byte 2/Byte 3, Byte 4/Byte 5 ... Byte 14/Byte 15, And Byte 16/Byte 17
8X305 PORT LOGIC
ADDRESS LATCHES AND DECODE LOGIC
8X305 DRIVERS/RECEIVERS
USER PORT ADDRESS DECODE
PROTECT BIT STATUS
USER PORT LOGIC
USER PORT DRIVERS/RECEIVERS
8
16
MCLK
SC
WC
IV0
IV1
IV2
IV3
IV4
IV5
IV6
IV7
19
21
22
23
24
25
26
27
28
29
30
31
2
12
13
14
15
16
17
18
19
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
D7A
D6A
D5A
D4A
D3A
D2A
D1A
D0A
D7B
D6B
D5B
D4B
D3B
D2B
D1B
D0B

Figure 1. Block Diagram of 8X320 Bus Interface Register Array

APRIL 1982

BUS INTERFACE REGISTER ARRAY 8X320



PIN NO.	PARAMETER		FUNCTION
1, 20	GND	Ground	Circuit ground.
2	$\overline{DMAE}$	Direct Memory Access Enable	Enables primary port to facilitate DMA transfers; does not affect secondary port.
3-18	$D0_A$-$D7_A$/ $D0_B$-$D7_B$	Primary Data Port	Sixteen 3-state lines used for data transfers to-and-from the primary data port; most significant bit is $D0_A$ and least significant bit is $D7_B$.
19	$\overline{ME}$	Master Enable	Enables secondary port when active low ($\overline{ME}$).
21	MCLK	Master Clock	When MCLK is high, and 8X320 is enabled ($\overline{ME}$ = Low), a register location may be either selected or written-into under control of SC and WC.
22	SC	Select Command	With SC high, WC low, MCLK high and $\overline{ME}$ low, data on $\overline{IV0}$ through $\overline{IV7}$ is interpreted as an address. If any one of the 16 register addresses (60_8-77_8) matches that on the I/O (IV) bus, that particular register is selected and remains selected until another address on the same bank (i.e. $\overline{ME}$ = low) is output on the I/O bus—at which time, the old register is deselected and a new register may or may not be selected.
23	WC	Write Command	With WC high, SC low, MCLK high, and $\overline{ME}$ low, the selected register stores contents of $\overline{IV0}$-$\overline{IV7}$ as data.
24-31	$\overline{IV0}$-$\overline{IV7}$	Secondary Data Port	Eight 3-state lines used to transfer data or I/O address to-and-from the secondary data port; most significant bit is $\overline{IV0}$ and least significant bit is $\overline{IV7}$.
32	WS	Write Strobe	When active high, data appearing at the primary port ($D0_A$-$D7_A$/$D0_B$-$D7_B$) is stored in the register array if the primary port is in the *write* mode.
33	R/$\overline{W}$	Read/Write Control	When this signal is high, primary port is in *read* mode; when signal is low, primary port is in *write* mode.
34	$\overline{PIOE}$	Programmed I/O Enable	When active low, primary port operates in programmed input/output mode with register to be read-from or written-into selected by A0-A3.
35-38	A0-A3	Primary Port Address Select	Selects register or register-pair that primary port is to read-from or write-into. Most significant bit is A3; least significant bit is A0.
39	B/$\overline{W}$	Byte/Word	When signal is high, the primary port operates in the byte (8-bit) mode; when signal is low, the primary port operates in the word (16-bit) mode.
40	V_{CC}	Power	+5 volts.

All barred symbols ($\overline{DMAE}$, etc.) denote signals that are asserted (or active) when low (logical 0); signals that are not barred are asserted in the high state (logical 1).

OPERATING CHARACTERISTICS

Memory Organization

Memory and address correlation for the 16 register array is shown in Figure 2. From the primary port, the sixteen 8-bit registers can be addressed in either (8-bit) or word (16-bit) format; in the word mode, the registers are addressed in pairs — $0_8/1_8$, $2_8/3_8$, $4_8/5_8$, $14_8/15_8$, and $16_8/17_8$. From the secondary port, all registers are addressed in byte format – 60_8 through 77_8. The memory consists of two 8-bit flag registers and fourteen 8-bit general-purpose registers. The flag registers facilitate information transfers between the two ports and, in addition, they protect certain registers from being written into from the primary port.

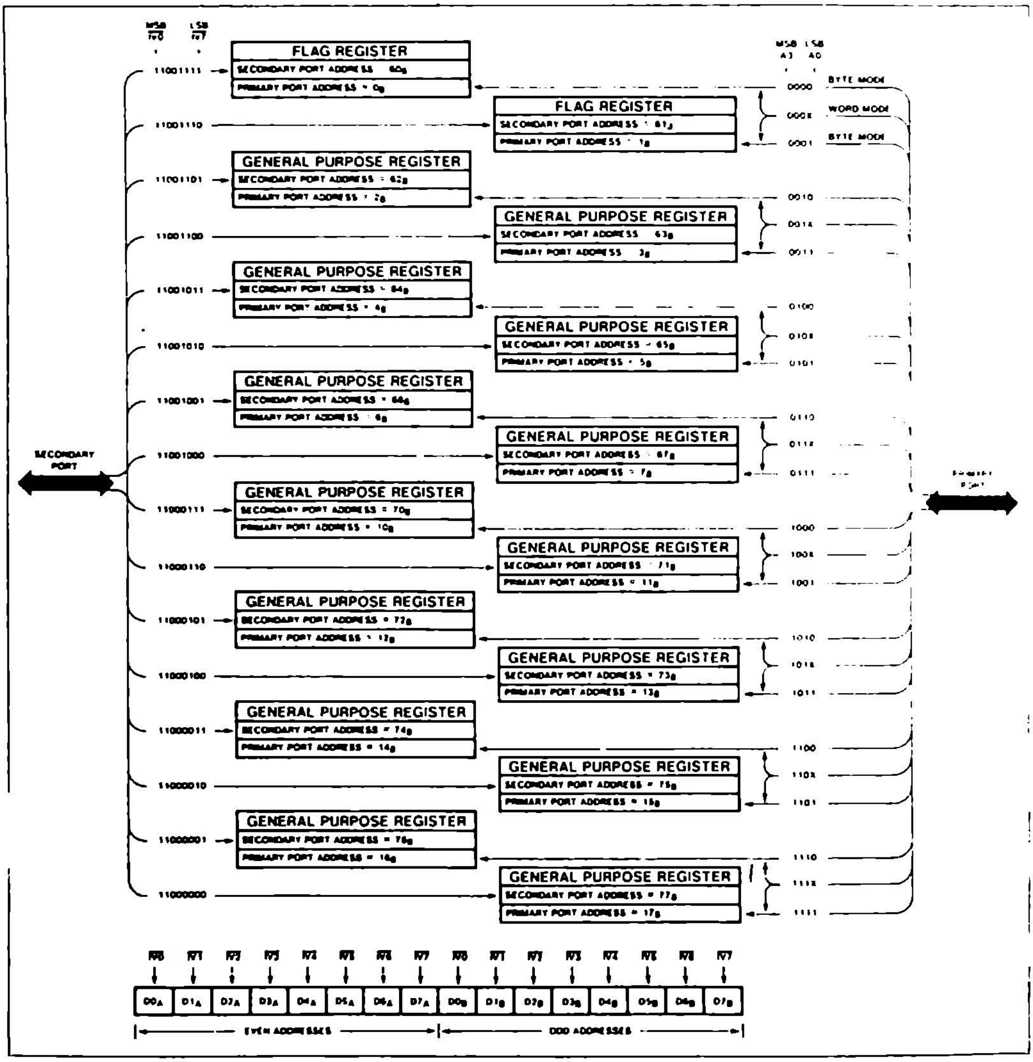

MSB IV0
LSB IV7
FLAG REGISTER
SECONDARY PORT ADDRESS 60₈
PRIMARY PORT ADDRESS = 0₈
FLAG REGISTER
SECONDARY PORT ADDRESS = 61₈
PRIMARY PORT ADDRESS = 1₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 62₈
PRIMARY PORT ADDRESS = 2₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS 63₈
PRIMARY PORT ADDRESS 3₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 64₈
PRIMARY PORT ADDRESS = 4₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 65₈
PRIMARY PORT ADDRESS = 5₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 66₈
PRIMARY PORT ADDRESS = 6₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 67₈
PRIMARY PORT ADDRESS = 7₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 70₈
PRIMARY PORT ADDRESS = 10₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 71₈
PRIMARY PORT ADDRESS = 11₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 72₈
PRIMARY PORT ADDRESS = 12₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 73₈
PRIMARY PORT ADDRESS = 13₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 74₈
PRIMARY PORT ADDRESS = 14₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 75₈
PRIMARY PORT ADDRESS = 15₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 76₈
PRIMARY PORT ADDRESS = 16₈
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 77₈
PRIMARY PORT ADDRESS = 17₈
11001111
11001110
11001101
11001100
11001011
11001010
11001001
11001000
11000111
11000110
11000101
11000100
11000011
11000010
11000001
11000000
SECONDARY PORT
MSB A3
LSB A0
BYTE MODE
WORD MODE
BYTE MODE
0000
000X
0001
0010
001X
0011
0100
010X
0101
0110
011X
0111
1000
100X
1001
1010
101X
1011
1100
110X
1101
1110
111X
1111
PRIMARY PORT
IV0 IV1 IV2 IV3 IV4 IV5 IV6 IV7 IV0 IV1 IV2 IV3 IV4 IV5 IV6 IV7
D0A D1A D2A D3A D4A D5A D6A D7A D0B D1B D2B D3B D4B D5B D6B D7B
EVEN ADDRESSES
ODD ADDRESSES

Figure 2. Memory and Address Organization for the 8X320

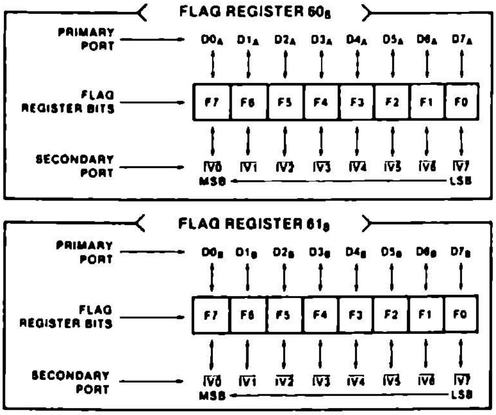
In either byte or word mode, the write-protect logic, implemented by bits F0 and F1 of register 60_8, inhibits the primary port from writing into addresses 16_8 and 17_8, respectively. Both write-protect bits (F0 and F1) can be read or written from the secondary port; the bits are read-only from the primary port.

As shown in Table 1, flag bits F2 through F7 of 60_8 and F0 through F7 of 61_8 are controlled by the fourteen general-purpose registers. When any one of these registers is written into by either port, the corresponding flag bit for that register is automatically set by internal logic of the 8x320. When information is read from any register, the corresponding flag bit must be reset by user software. Except for the write-protect bits, all other flag bits can be read or reset from the primary or the secondary port. Table 2 shows the relationship between bits of the flag registers and bits of the primary and secondary ports.

Table 1. Control of the Two Flag Registers

Flag Registers	60_8 (0_8)	F2	F3	F4	F5	F6	F7								
	61_8 (1_8)							F0	F1	F2	F3	F4	F5	F6	F7
Octal Address of Controlling Byte	Primary	2	3	4	5	6	7	10	11	12	13	14	15	16	17
	Secondary	62	63	64	65	66	67	70	71	72	73	74	75	76	77

Table 2. Relationship Between Flag Register Bits and Those of Primary and Secondary Ports



FUNCTION AND CONTROL OF PRIMARY PORT

The primary port provides an 8-bit (byte) or 16-bit (word) interface between the 16-byte memory and the user's host system. If the host is an 8-bit system (or 16-bit system operating in Byte mode), the sixteen bidirectional I/O lines must be tied together ($D0_A$ to $D0_B$, $D1_A$ to $D1_B$, . . . and $D7_A$ to $D7_B$); when data is input or output on $D0_A$ through $D7_A$, the remaining eight lines ($D0_B$ through $D7_B$) are high-Z and vice-versa.

Other than the Byte/$\overline{\text{Word}}$ control line, specific operating characteristics of the primary port are controlled by two signals—$\overline{\text{PIOE}}$ (Programmed I/O Enable) and $\overline{\text{DMAE}}$ (Direct Memory Access Enable). When $\overline{\text{PIOE}}$ is active (low) and $\overline{\text{DMAE}}$ is inactive (high), the primary port operates in the programmed I/O mode—refer to Table 3. In this mode of operation, the register to be read-from or written-into is determined by four address lines (A0 through A3) and the Byte/$\overline{\text{Word}}$ control line—see Figure 2 and Table 4. In the DMA mode of operation, A1, A2, and A3 are not used; data is read-from or written-into preassigned registers: bytes 16_8 (76_8) and 17_8 (77_8) for the byte mode of operation and bytes 14_8 (74_8)/15_8 (75_8) and 16_8 (76_8)/17_8 (77_8) for the word mode of operation. In both cases, switching between bytes 16_8 and 17_8 in the byte mode and 14_8/15_8 and 16_8/17_8 in the word mode is controlled by A0 (the least significant address bit). Refer to Table 5.

Table 3. Mode Control of Primary Port

MODE	$\overline{\text{PIOE}}$	$\overline{\text{DMAE}}$
Disabled (output)	1	1
Programmed I/O	0	1
DMA	X	0

X = Don't Care

Table 4 defines programmed I/O operation of the primary port in terms of read/write functions and Byte/$\overline{\text{Word}}$ control. In the byte mode, data is read-from or written-into the even addresses (0_8, 2_8, 4_8, 6_8, 10_8, 12_8, 14_8, and 16_8) via data lines $D0_A$ through $D7_A$; data is read-from or written-into odd addresses (1_8, 3_8, 5_8, 7_8, 11_8, 13_8, 15_8, and 17_8) via data lines $D0_B$ through $D7_B$. When A0 is low (logical 0), even addresses are selected and when A0 is high (logical 1), odd addresses are selected; thus, A0 is the LSB of a 4-bit address. In the word mode, the state of A0 is irrelevant, since both the odd and even bytes are, simultaneously, read-from or written-into; thus, a register pair is selected by a 3-bit address, A1 being the LSB.

In the DMA mode of operation with $\overline{\text{DMAE}}$ set to 0 and other conditions satisfied, data is directly transferred to-or-from specified memory locations under control of Byte/$\overline{\text{Word}}$, R/$\overline{\text{W}}$, and A0. The state of the Byte/$\overline{\text{Word}}$ control line determines whether the data word is 8 bits or 16 bits. The A0 address line correlates eight of

Table 4. Primary Port Operating in Programmed I/O Mode

MODE	B/$\overline{\text{W}}$	A0	$D0_A$-$D7_A$ (Even Addresses)	$D0_B$-$D7_B$ (Odd Addresses)
Read	0 (Word)	X	Stored Data	Stored Data
Read	1 (Byte)	0	Stored Data	Hi-Z
Read	1 (Byte)	1	Hi-Z	Stored Data
Write	0 (Word)	X	Write	Write
Write	1 (Byte)	0	Write	No Change
Write	1 (Byte)	1	No Change	Write

X = Don't Care

the sixteen data lines ($D0_A$-$D7_A$ or $D0_B$-$D7_B$) with the proper byte/word location. Thus, in the word mode, the exchange of data between the memory and the primary port occurs via $D0_A$-$D7_A$ for bytes 14_8 and 16_8 and via $D0_B$-$D7_B$ for bytes 15_8 and 17_8. The byte mode of operation is similar, except that the unused eight lines are three-stated.

FUNCTION AND CONTROL OF SECONDARY PORT

The secondary port provides an 8-bit interface between the sixteen memory registers and the 8X305 (or other processor). As shown in Table 6, the secondary-port interface is controlled by five input signals and a status latch. The status latch is set when SC is high (MCLK high/$\overline{ME}$ low) and a valid memory address (60_8-77_8) is presented to the 8X320 via the secondary data port ($\overline{IV0}$-$\overline{IV7}$). The latch is cleared by internal logic when an invalid memory address is presented at the secondary port. In all read/write operations from the secondary port, the status latch acts like a master enable; data can be transferred only if the status latch is set.

Table 5. DMA Operation of the Primary Port

MODE	BYTE/WORD	A0	$D0_A$-$D7_A$	$D0_B$-$D7_B$
Read	0 (Word)	0	Data stored in byte 14_8	Data stored in byte 15_8
Read	0 (Word)	1	Data stored in byte 16_8	Data stored in byte 17_8
Read	1 (Byte)	0	Data stored in byte 16_8	HI-Z
Read	1 (Byte)	1	HI-Z	Data stored in byte 17_8
Write	0 (Word)	0	Write to byte 14_8	Write to byte 15_8
Write	0 (Word)	1	Write to byte 16_8	Write to byte 17_8
Write	1 (Byte)	0	Write to byte 16_8	HI-Z
Write	1 (Byte)	1	HI-Z	Write to byte 17_8

Table 6. Functional Control of Secondary Port

$\overline{ME}$	SC¹	WC¹	MCLK	$R/\overline{W}$	STATUS LATCH	FUNCTION OF SECONDARY BUS
L	L	L	X	X	Set	Output data from 8X320 memory to 8X305
L	L	H	H	H	Set	Data from 8X305 is input and written into a previously selected memory location of the 8X320 (Note 2)
L	L	H	H	L	Set	With the primary port in the write mode ($R/\overline{W}$ = 0), the secondary port is overridden and cannot write to the same register addressed by the primary port; however, the register addressed by the primary port can be read and any other register can be read-from or written-into from the secondary port (Note 2).
L	H	L	H	X	X	Data transmitted to the secondary port via the $\overline{IV}$ bus is interpreted as an address; if address is within range of 60_8-77_8 the memory status latch is subsequently set.
L	L	H	L	X	X	Inactive
L	H	L	L	X	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

Notes

1. The SC and WC lines should never both be high at the same time. The 8X305 processor never generates this condition.
2. During read or write operations, the same register can be simultaneously addressed from either port. For any write operation by both ports on the same register, the primary port has priority; other than this, the 8X320 does not indicate error conditions or resolve conflicts.
3. X = Don't Care

BUS INTERFACE REGISTER ARRAY 8X320

DC CHARACTERISTICS $0°C \leq T_A \leq 70°C, 4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1, 2]	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5	5.25	V
$V_{IN (L)}$	Low level input voltage				0.80	V
$V_{IN (H)}$	High level input voltage		2.0			V
V_{OL}	Low level output voltage	$V_{CC} = 4.75V, I_{OL} = 16mA$			0.55	V
V_{OH}	High level output voltage	$V_{CC} = 4.75V, I_{OH} = -3mA$	2.40			V
V_{CL}	Input clamp voltage	$I_I = -5mA$			−1.00	V
I_{CC}	Supply current	V_{CC} = 5.25V (Both ports high-Z)			270	mA
I_{OS}	Short circuit output current[3]	$V_{CC} = 4.75V$	−20		−100	mA
$I_{IN (L)}$	WC, MCLK, SC, & $\overline{ME}$	$V_{CC} = 5.25V, V_{IL} = 0.50V$			−1.0	mA
$I_{IN (L)}$	B/$\overline{W}$	$V_{CC} = 5.25V, V_{IL} = 0.50V$			−1.6	mA
$I_{IN (L)}$	A0-A3	$V_{CC} = 5.25V, V_{IL} = 0.50V$			−1.0	mA
$I_{IN (L)}$	$\overline{DMAE}$	$V_{CC} = 5.25V, V_{IL} = 0.5V$			−800	μA
$I_{IN (L)}$	WS, $\overline{PIOE}$, & R/$\overline{W}$	$V_{CC} = 5.25V, V_{IL} = 0.5V$			−400	μA
$I_{IN (L)}$	$\overline{IV0}$-$\overline{IV7}$	$V_{CC} = 5.25V, V_{IL} = 0.5V$			−400 each line	μA
$I_{IN (L)}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$	$V_{CC} = 5.25V, V_{IL} = 0.5V$			−400 each line	μA
$I_{IN (H)}$	WC, SC, MCLK, & $\overline{ME}$	$V_{CC} = 5.25V, V_{IH} = 5.25V$			100	μA
$I_{IN (H)}$	B/$\overline{W}$	$V_{CC} = 5.25V, V_{IH} = 5.25V$			240	μA
$I_{IN (H)}$	A0	$V_{CC} = 5.25V, V_{IH} = 5.25V$			120	μA
$I_{IN (H)}$	A1-A3	$V_{CC} = 5.25V, V_{IH} = 5.25V$			60	μA
$I_{IN (H)}$	$\overline{DMAE}$	$V_{CC} = 5.25V, V_{IH} = 5.25V$			120	μA
$I_{IN (H)}$	WS, $\overline{PIOE}$, & R/$\overline{W}$	$V_{CC} = 5.25V, V_{IH} = 5.25V$			60	μA
$I_{IN (H)}$	$\overline{IV0}$-$\overline{IV7}$ and $D0_A$-$D7_A$/$D0_B$-$D7_B$	$V_{CC} = 5.25V, V_{IH} = 5.25V$			100	μA

Notes

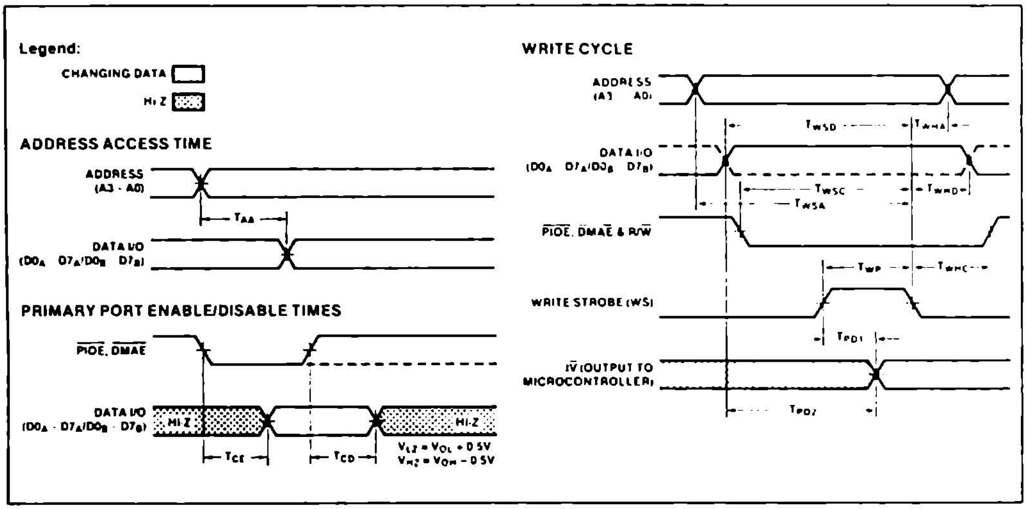
[1] Operating temperature ranges are guaranteed after terminal equilibrium has been reached

[2] All voltages are measured with respect to ground terminal

[3] Short only one output at a time

AC CHARACTERISTICS OF PRIMARY PORT

$0°C \le T_A \le 70°C$, $4.75V \le V_{CC} \le 5.25V$
Loading See Test Circuit


Legend:
CHANGING DATA
HI Z
ADDRESS ACCESS TIME
ADDRESS (A3 - A0)
T_{AA}
DATA I/O ($D0_A$ - $D7_A$/$D0_B$ - $D7_B$)
PRIMARY PORT ENABLE/DISABLE TIMES
$\overline{PIOE}$, $\overline{DMAE}$
DATA I/O ($D0_A$ - $D7_A$/$D0_B$ - $D7_B$)
HI-Z
HI-Z
T_{CE}
T_{CD}
$V_{LZ} = V_{OL} + 0.5V$
$V_{HZ} = V_{OH} - 0.5V$
WRITE CYCLE
ADDRESS (A3 - A0)
T_{WSD}
T_{WHA}
DATA I/O ($D0_A$ - $D7_A$/$D0_B$ - $D7_B$)
T_{WSC}
T_{WHD}
T_{WSA}
$\overline{PIOE}$, $\overline{DMAE}$ & R/$\overline{W}$
T_{WP}
T_{WHC}
WRITE STROBE (WS)
T_{PD1}
$\overline{IV}$ (OUTPUT TO MICROCONTROLLER)
T_{PD2}

PARAMETER		FROM	TO	LIMITS			UNIT
				Min	Typ	Max	
T_{AA}	Address Access Time	A3-A0	$D0_A$-$D7_A$/$D0_B$-$D7_B$			45	ns
T_{CE}	Primary port enable time	↓$\overline{PIOE}$ ↓$\overline{DMAE}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$			30	ns
T_{CD}	Primary port disable time	↑$\overline{PIOE}$ ↑$\overline{DMAE}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$			35	ns
T_{WSA}	Address setup time	A3-A0	↓WS	40			ns
T_{WHA}	Address hold time	↓WS	A3-A0	0			ns
T_{WSD}	Primary port data setup time	$D0_A$-$D7_A$/$D0_B$-$D7_B$	↓WS	30			ns
T_{WHD}	Primary port data hold time	↓WS	$D0_A$-$D7_A$/$D0_B$-$D7_B$	0			ns
T_{WSC}	Write mode control setup time	$\overline{PIOE}$ $\overline{DMAE}$ R/$\overline{W}$	↓WS ↓WS ↓WS	30 40 30			ns ns ns
T_{WHC}	Write mode control hold time	↓WS	$\overline{PIOE}$ $\overline{DMAE}$ R/$\overline{W}$	10 10 10			ns ns ns
T_{WP}	Write strobe pulse width			25			ns
T_{PD1}[1]	Primary port data delay	$D0_A$-$D7_A$/$D0_B$-$D7_B$	$\overline{IV0}$-$\overline{IV7}$			75	ns
T_{PD2}[2]	Primary port data delay from WS	↑WS	$\overline{IV0}$-$\overline{IV7}$			75	ns

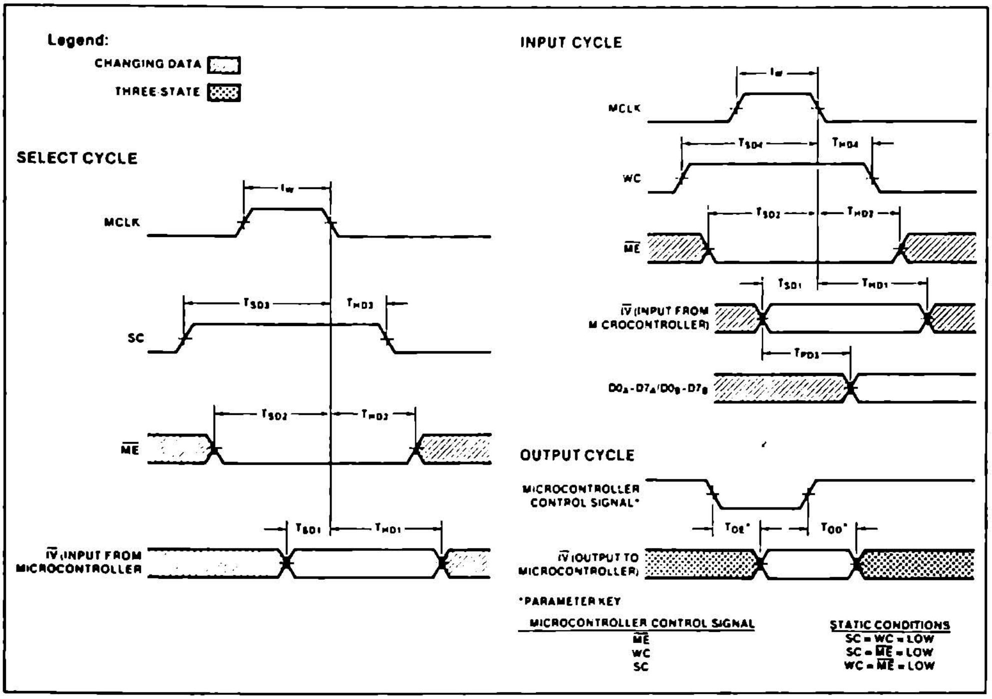
Notes

[1] Measurement with Write Strobe set High and the control signals of the secondary port set for output data from the same register

[2] Measurement with primary port data stable and control signals of secondary port set for output data from the same register

AC CHARACTERISTICS OF SECONDARY PORT

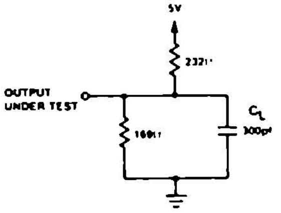
0°C ≤ T_A ≤ 70°C, 4.75V ≤ V_{CC} ≤ 5.25V
Loading: See Test Circuit


Legend:
CHANGING DATA
THREE-STATE
SELECT CYCLE
MCLK
tw
SC
TSD3
THD3
ME
TSD2
THD2
TSD1
THD1
IV (INPUT FROM MICROCONTROLLER)
INPUT CYCLE
MCLK
tw
WC
TSD4
THD4
ME
TSD2
THD2
IV (INPUT FROM MICROCONTROLLER)
TSD1
THD1
TPD3
D0A-D7A/D0B-D7B
OUTPUT CYCLE
MICROCONTROLLER CONTROL SIGNAL*
TOE*
TOD*
IV (OUTPUT TO MICROCONTROLLER)
*PARAMETER KEY
MICROCONTROLLER CONTROL SIGNAL
ME
WC
SC
STATIC CONDITIONS
SC = WC = LOW
SC = ME = LOW
WC = ME = LOW

PARAMETER		FROM	TO	LIMITS			UNIT
				Min	Typ	Max	
t_W	MCLK pulse width			30			ns
T_{SD1}	Data setup time	$\overline{\text{IV0}}$-$\overline{\text{IV7}}$	↓MCLK	35			ns
T_{SD2}	$\overline{\text{ME}}$ setup time	$\overline{\text{ME}}$	↓MCLK	30			ns
T_{SD3}	SC setup time	SC	↓MCLK	30			ns
T_{SD4}	WC setup time	WC	↓MCLK	30			ns
T_{HD1}	Data hold time	↓MCLK	$\overline{\text{IV0}}$-$\overline{\text{IV7}}$	0			ns
T_{HD2}	$\overline{\text{ME}}$ hold time	↓MCLK	$\overline{\text{ME}}$	0			ns
T_{HD3}	SC hold time	↓MCLK	SC	0			ns
T_{HD4}	WC hold time	↓MCLK	WC	0			ns
T_{PD3} (Note)	$\overline{\text{IV}}$ propagation delay	$\overline{\text{IV}}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$			45	ns
T_{OE}	Output enable	$\overline{\text{ME}}$, SC, or WC	$\overline{\text{IV0}}$-$\overline{\text{IV7}}$			30	ns
T_{OD}	Output disable	$\overline{\text{ME}}$, SC, or WC	$\overline{\text{IV0}}$-$\overline{\text{IV7}}$			20	ns

Note
Measured with MCLK = High and control signals of the primary port set for output data from the same register

TEST CIRCUIT


5V
232Ω
OUTPUT
UNDER TEST
160Ω
C_L
300pF

Note
C_L includes jig and probe capacitance

PACKAGE DATA:

Type Plastic or Ceramic
Configuration DIP
Width 0.6 inches
Length 2.0 inches
Pin Centers 100 mils

ORDERING INFORMATION:

N8X320N (Plastic)
N8X320I (Ceramic)

ARCHITECTURAL OVERVIEW

The Signetics 8X330 Floppy Disk Formatter/Controller (Figure 1) is a monolithic peripheral device of the 8X300 family. The chip uses Bipolar-Schottky/I²L-Technology and some very unique features to provide 8X330 customers with a competitive edge in both simple and complicated disk-controller designs. The competitive advantage is measurable in terms of "systems parts count", "error-correction capabilities", and "overall design concepts" that are applications oriented. Except for a crystal, a capacitor, an external transistor acting as a series-pass element for the on-chip voltage regulator, an active low-pass filter, and an optional off-chip voltage controlled oscillator (refer to Features and Option), the 8X330 contains all processing circuits and the required control logic to encode/decode double-density (MFM/M²FM) and single-density (FM) codes. Even the data-separation and write-precompensation logic are located on the chip, in addition, 16-bytes of scratch-pad RAM are provided for storage of various control/status parameters.

FEATURES

- Single or double density encoding/decoding
- On-chip data separator
- Programmable:
  FM, MFM, and M²FM encoding/decoding
  Preamble Polarity
  Data transfer rate
  Address mark encoding/decoding
  Sector length
  Output port (7-bits disk command)
  Input port (5-bits disk status)
- Write Precompensation with on/off control
- On-chip phase lock loop
- CRC generator with software-controlled error correction capabilities
- 40-pin package
- +5 volt operation

OPTION External Voltage Controlled Oscillator (VCO) For critical applications, window margins can be improved by as much as 6% with the use of an external VCO

NOTES
1 Components shown with dotted lines are located off-chip
2 Certain bits of the status/control registers connected to the I/O bus are read only; this distinction is not shown in the diagram
3 Greater detail for the more complicated blocks (Phase-Lock Loop, Data Separator, etc) are shown in other areas of this data sheet

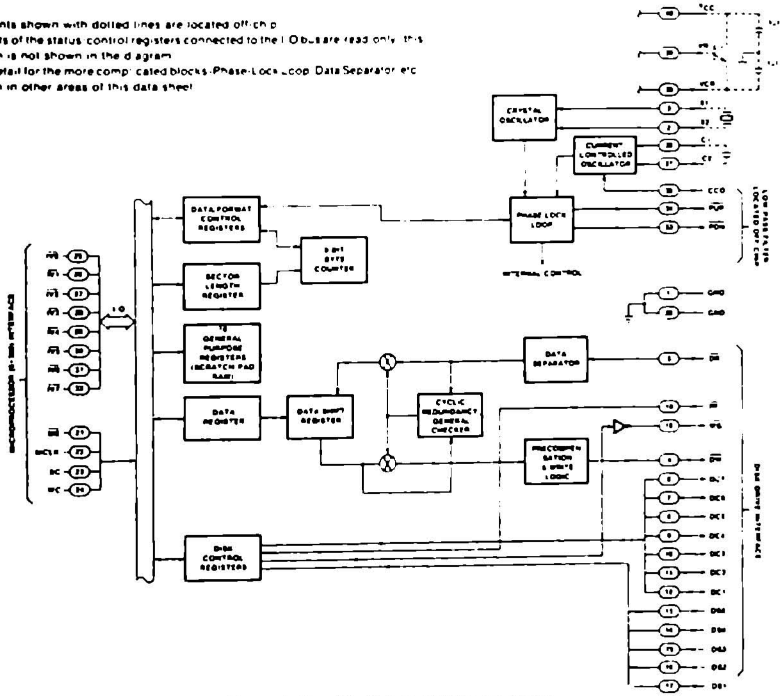
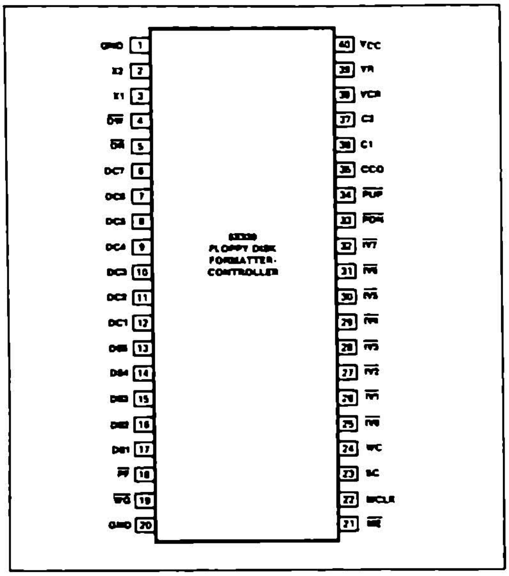

VCC
VCO
CRYSTAL OSCILLATOR
CURRENT CONTROLLED OSCILLATOR
CCO
PHASE LOCK LOOP
INTERNAL CONTROL
LOW PASS FILTER LOCATED OFF CHIP
GND
GND
DATA FORMAT CONTROL REGISTERS
8-BIT BYTE COUNTER
SECTOR LENGTH REGISTER
16 GENERAL PURPOSE REGISTERS (SCRATCH PAD RAM)
DATA REGISTER
DATA SHIFT REGISTER
DISK CONTROL REGISTERS
I/O
MICROPROCESSOR (8-BIT) INTERFACE
CYCLIC REDUNDANCY GENERAL CHECKER
DATA SEPARATOR
PRECOMPENSATION & WRITE LOGIC
DISK DRIVE INTERFACE

Figure 1. Simplified Block Diagram of 8X330

8X330 PACKAGE/PIN DESIGNATIONS


GND
X2
X1
$\overline{DW}$
$\overline{DR}$
DC7
DC6
DC5
DC4
DC3
DC2
DC1
DS5
DS4
DS3
DS2
DS1
$\overline{PF}$
$\overline{WG}$
GND
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
8X330
FLOPPY DISK
FORMATTER-
CONTROLLER
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
VCC
VR
VCR
C2
C1
CCO
$\overline{PUP}$
$\overline{PDN}$
$\overline{IV7}$
$\overline{IV6}$
$\overline{IV5}$
$\overline{IV4}$
$\overline{IV3}$
$\overline{IV2}$
$\overline{IV1}$
$\overline{IV0}$
WC
SC
MCLK
$\overline{ME}$

PIN NO.	MNEMONIC & DEFINITION		FUNCTION
1, 20	GND	Ground	Circuit ground
2, 3	X1, X2	Crystal inputs	Inputs from a crystal that determines frequency of an on-chip crystal oscillator
4	$\overline{DW}$	Data write	A series of negative-going pulses transmitted to the disk drive. The data write signal produces pulses (with precompensation, if required) for data and clock in accordance with the applicable encoding rules (FM, MFM or M²FM)
5	$\overline{DR}$	Data read	Negative-going pulses transmitted from the disk drive to a Schmitt-trigger input of the 8X330; these pulses represent encoded data and clock from the disk media
6-12	DC1-DC7	Disk commands	Seven outputs from the 8X330 that allow general-purpose control, of one or more disk drives
13-17	DS1-DS5	Disk status	Five general-purpose Schmitt-trigger inputs from the disk drive (or drives) that provide status information for the 8X300
18	$\overline{PF}$	Power fail	Schmitt-trigger input from external logic that is active (low) when the "user-sensed" power supply voltage drops below a predetermined value.
19	$\overline{WG}$	Write gate	When active (low), this 40-milliampere open-collector output enables writing to the disk media. When $\overline{PF}$ is low, the write gate is inhibited during periods of power supply uncertainty
21	$\overline{ME}$	Master enable	When this input signal is active (low), the 8X330 can be accessed and enabled by the 8X300. Refer to the $\overline{LB}$ and $\overline{RB}$ pinout discriptions of the 8X300 for further details
22	MCLK	Master clock	When active high and with $\overline{ME}$ in the active-low state, this input signal provides a means whereby the I/O output from the 8X300 is interpreted as an enabling address (provided there is an address match) or as input data if one of the 8X330 registers has already been selected)
23	SC	Select command	When this signal is active high the information output on pins $\overline{IV0}$-$\overline{IV7}$ of the 8X300 is interpreted as an address input by the 8X330
24	WC	Write command	When this signal is active high the information output on pins $\overline{IV0}$-$\overline{IV7}$ of the 8X300 is interpreted as input data by the 8X330
25-31	$\overline{IV0}$-$\overline{IV7}$	Input/output lines	Eight three-state input/output lines that provide bidirectional data transfers between the 8X300 and the enabled I/O device. $\overline{IV7}$ is the *Least Significant Bit*
33	$\overline{PDN}$	Pump down output	Open-collector output of on-chip phase detector which indicates by a negative-going quantized pulse-width modulated signal that internal CCO frequency is too high
34	$\overline{PUP}$	Pump up output	Open-collector output of on-chip phase detector which indicates by a negative-going, quantized pulse-width modulated signal that internal CCO frequency is too low
35	CCO	Frequency Control Input for Current-Controlled Oscillator	Variable input current from external low-pass filter that controls the frequency of the oscillator
36-37	C1, C2	Capacitor input terminals	Inputs for capacitor that determines center frequency of the current-controlled oscillator
38	VCR	Regulated supply voltage	DC voltage input from emitter of external series-pass transistor, this voltage powers internal logic of chip
39	VR	Reference voltage	Reference voltage output to base of series-pass transistor, this reference controls VCR
40	Vcc	Supply voltage	+5 volt power

SYSTEM INTERFACE

A typical floppy disk controller using an 8X300 microcontroller and the 8X330 is shown in Figure 2. The non-shaded portion of this particular configuration can service the command, status, and input/output requirements of two double-sided disk drives and, under software supervision, the system can read/write single-density (FM) or double-density (MFM/M²FM) codes. Interface requirements are simple—on one hand, consisting of the 8X300 microcontroller and, on the other, the two disk drives. All 8X330 control and data registers directly linked to the microprocessor Interface (Figure 1) are addressable and appear to the 8X300 as simple I/O ports; a 13-bit address bus and a 16-bit instruction bus provide communications between the 8X300 and up to 8K of microprogram storage.

The disk-drive interface consists of seven (7) output control lines (DC1-DC7), five (5) input status lines (DS1-DS5), a write gate ($\overline{WG}$), a data- write output ($\overline{DW}$), and a data-read input ($\overline{DR}$). The twelve command/status lines are not dedicated; thus, the user can assign system functions to best suit a given application. As shown in Figure 2, all control lines except $\overline{WG}$ are buffered to accommodate a reasonable distance between the controller and the disk media, the Write Gate, being a 40-milliampere output, requires no buffering.

As shown by the shaded part of Figure 2, the control and status lines can be expanded with peripheral hardware—the 8T32 (in this example) being only one method of implementation. Using this particular technique, one I/O port is totally dedicated to output control, whereas, the other port is totally dedicated to input status. With additional hardware and supporting software, the disk-drive system can be expanded without limit; however, from a point of being practical, five or six drives is sufficient for most applications. By using the programmable features of the 8X330, the user can emphasize and prioritize those system parameters that are most important—economics, reliability and/or speed.

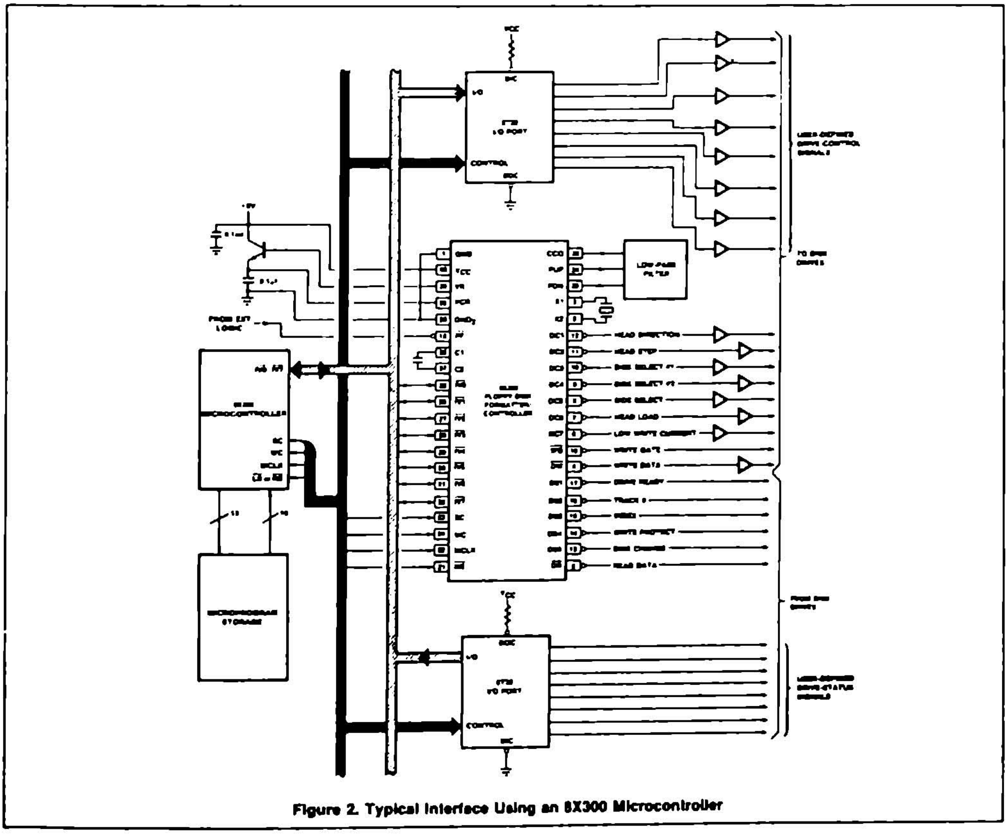


Figure 2. Typical Interface Using an 8X300 Microcontroller

FUNCTIONAL OPERATION

As shown in Figure 2, the interface between the 8X300 and 8X330 consists of twelve (12) lines—$\overline{IV0}$-$\overline{IV7}$, SC, WC, MCLK, and $\overline{ME}$; the Master Enable ($\overline{ME}$) input (pin 21) is driven from either the $\overline{LB}$ (Left Bank) or $\overline{RB}$ (Right Bank) output of the 8X300. An expanded view of this interface is shown in Figure 3 and, as indicated, the 8X330 appears as a number of addressable registers (110_8-127_8 and 132_8-137_8) under input/output control of the 8X300. These registers are used for general-purpose storage, data-transfer operations, disk commands, disk status, and various control functions. Design-oriented information for these registers and other data-processing/logic functions of the 8X330 are described in the paragraphs that follow; in all of these registers, bit 0 is the Most Significant Bit (MSB).

NOTE
When power is first applied to the 8X330, the Disk Command lines (DC1-DC7), the Write Gate ($\overline{WG}$) output, and contents of Command/Status Register #2 (CSR #2) are set to 1 (high). The wakeup state of all other bits is undefined.

General-Purpose Register File

This general-purpose (scratch pad) memory is directly accessible by the 8X300 and is used to store system variables such as track address, sector address and other necessary parameters. The sixteen 8-bit registers (110_8-127_8) provide sufficient on-chip memory to accommodate a minimum of two disk drives; the maximum number of drives that this non-dedicated memory file can support depends on several factors—system configuration, reliability requirements, economic constraints, and so on. Because of the on-chip file, all other system memory can be dedicated to the purpose of handling data to-and-from the disk med a

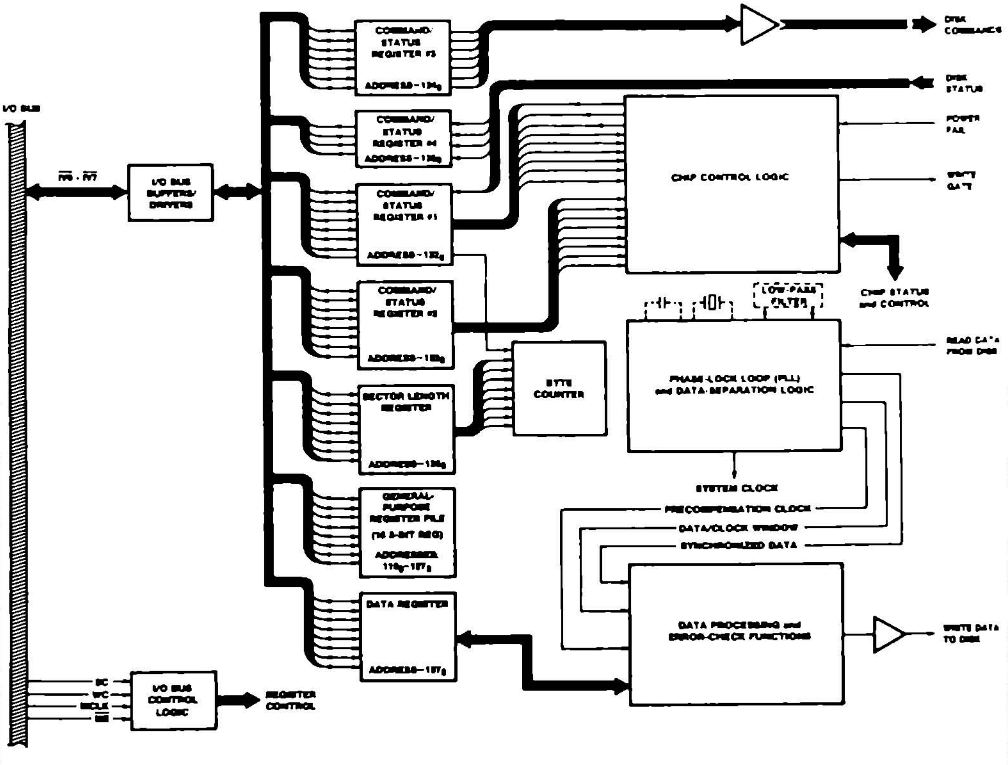

I/O BUS
$\overline{IV0}$ - $\overline{IV7}$
I/O BUS BUFFERS/ DRIVERS
COMMAND/ STATUS REGISTER #3
ADDRESS-134_8
COMMAND/ STATUS REGISTER #4
ADDRESS-135_8
COMMAND/ STATUS REGISTER #1
ADDRESS-132_8
COMMAND/ STATUS REGISTER #2
ADDRESS-133_8
SECTOR LENGTH REGISTER
ADDRESS-136_8
GENERAL-PURPOSE REGISTER FILE (16 8-BIT REG)
ADDRESSES 110_8-127_8
DATA REGISTER
ADDRESS-137_8
SC
WC
MCLK
$\overline{ME}$
I/O BUS CONTROL LOGIC
REGISTER CONTROL
DISK COMMANDS
DISK STATUS
POWER FAIL
WRITE GATE
CHIP CONTROL LOGIC
CHIP STATUS and CONTROL
LOW-PASS FILTER
BYTE COUNTER
PHASE-LOCK LOOP (PLL) and DATA-SEPARATION LOGIC
READ DATA FROM DISK
SYSTEM CLOCK
PRECOMPENSATION CLOCK
DATA/CLOCK WINDOW
SYNCHRONIZED DATA
DATA PROCESSING and ERROR-CHECK FUNCTIONS
WRITE DATA TO DISK

Figure 3. An Expanded View of the 8X330/8X300 Interface

Command/Status Register #1 (CSR 1/Address 132$_8$)
The disk status (read) or disk-command (write) contents of this register are interpreted as follows, unless otherwise indicated, all bits of CSR 1 are read/write from the I/O bus

Bit 0 (Write Gate Enable)
Enables write gate output ($\overline{WG}$/pin 19) to disk drive(s)—the write gate ($\overline{WG}$) cannot be enabled unless the PF input pin (18) is high. When the WGE bit is set to 0, the $\overline{WG}$ ouput pin is low (enabled); when WGE is set to 1, the $\overline{WG}$ output is high (disabled). If the PF input goes low while the $\overline{WG}$ output is low, the $\overline{WG}$ output will go high and the Write Gate Enable bit is reset to 1.

Bit 1 (CRC Enable)
When set to 1, permits internal CRC register to compute remainders on the data stream in either read or write modes of operation. When set to 0, the CRC register becomes the source of data. A change in the CRC Enable bit does not become effective until the "next BYTRA flag appears" following the command bit change—refer to description of CSR 1/Bit 6.

Bit 2 (Data Register Control)
When set to 0, contents of data register consists of interleaved data-and-clock bits, starting from the MSB ($\overline{IV0}$) position, register contents are: Clock 1, Data 1, Clock 2, Data 2, Clock 3, Data 3, Clock 4, and Data 4. When writing an address mark, the appropriate data/clock pattern is loaded into the data register by the 8X300. Since each byte of data from the processor becomes an interleaved pattern (4-bits of data and 4-bits of clock) in the 8X330 data register, two bytes from the processor are required to write each full byte of address mark to the drive—eight bit cells with each cell containing a possible data and/or clock transition, or a total of 16 bit positions. When writing address marks, the normal on-chip clock insertion circuitry of the 8X330 is inhibited, thus, the user is free to define any clock/data pattern for the address mark.

When reading address marks, the data register is loaded with data and clock representing four bit cells from the disk media. The information in the data register can then be compared with the expected address mark by the 8X300 on a nibble-by-nibble basis. When the DRC bit is set to 1, the data register contains separated data (no clocks). A state change in this bit does not become effective until the "next BYTRA flag appears" following the state-change command.

Bit 3 (Sync Enable)
The Sync Enable bit allows the on-chip data separator to obtain bit and byte synchronization; this bit also controls initialization of the CRC Register. With the 8X330 in Read mode and with Bit 3 set to 0, bit synchronization occurs. The Preamble field is assumed to be all "zeroes" or all "ones" as determined by the Preamble Select bit (CSR 2/Bit 4).

When the proper number of preamble bytes, as determined by the disk-control program, have been found, the Sync Enable bit should be changed, under program direction, to a 1. This puts the 8X330 in the Address-Mark search mode. Accordingly, all bits of the CRC Register are preset to 1, the BYte TRAnsfer flag is inhibited and the 8X330 examines the data stream for an Address Mark. The Address Mark is detected by observing the data and clock bits to find a change in the normal Preamble pattern. Byte synchronization is achieved by assuming that the change occurred in the bit cell determined by two Bit Select bits (CSR 2/Bits 2 and 3).

When the pattern change is found indicating the start of an Address Mark, the 8X330 starts CRC computation and synchronizes BYTRA to the byte boundaries. Note that the 8X330 presumes an Address Mark by finding a change in the preamble pattern, however, it is up to the 8X300 to read the Address Mark and to establish its validity or non-validity.

In write mode, setting the Sync Enable bit to 0 presets all bits of the CRC Register to 1. Setting the Sync Enable bit to 1 allows CRC computation to begin at the next byte boundary.

Bit 4 (Load Counter)
When set to 1, transfers 8-bits of data from Sector Length register and 1-bit (MSB) of data from Byte Counter (refer to next description) to 9-bit Byte Counter. Loading of the 9-bit Byte Counter is effective one bit-cell time after the Load Counter bit is set to 1. In both the read and write modes of operation, the Byte Counter is incremented by BYTRA. The Load Counter bit is self-clearing and always returns a 0 when read.

NOTE
The Load Counter bit must be set one or more instruction cycles after setting the Byte Counter MSB that is bits 4 and 5 of CSR 1 cannot be set during the same instruction cycle

Bit 5 (Byte Counter MSB)
This bit is used to set and monitor the state of the ninth (MSB) bit in the Byte Counter, reading this bit always returns the current state of MSB in the Byte Counter. The MSB of the Byte Counter is set to the value of CSR 1/Bit 5 when the Load Counter bit (CSR 1/Bit 4) is asserted—refer to preceding description

NOTE
The Byte Counter MSB must be set one or more instruction cycles before the Load Counter bits—bits 4 and 5 of CSR 1 cannot be set during the same instruction cycle

Bit 6 (BYTRA)
During a disk read operation, the BYte TRAnsfer flag is automatically set to 0 when 8-bits of information are transferred from the Data Shift Register to the Data Register—see Figure 1. During a disk write operation, BYTRA is automatically set to 0 when 8-bits are transferred from the Data Register to the Data Shift Register. BYTRA (a read-

only bit) is reset to a 1 when the Data Register (address 137_8) is selected by the user's program. During read/write operations, the 1-to-0 transition of the BYTRA flag increments the Byte Counter to keep count of bytes read or bytes written. All read-only bits of the 8X330 are designed to remain stable during the monitor period; thus, to read a status change of BYTRA, Disk-Status bit, the Byte Counter MSB, or other read-only bit requires a two-instruction loop similar to:

```
TEST     SEL     CSR 1
         NZT     BYTRA,    TEST
```

Bit 7 (Disk Status 1)
Reflects state (0 or 1) of input DS1 (pin 17); this is a user-definable read-only bit.

NOTE
A high input on any one of the Disk Status lines of the 8X330 is read by the 8X300 program as a logical 1 and a low input on the status lines is read as a logical 0

Command/Status Register #2 (CSR 2/Address 133_8)

The disk status (read) or disk-command (write) contents of this register are interpreted as follows:

Bit 0 (Precompensation Enable)
This command bit determines whether or not precompensation is applied to the data stream being written onto the disk. When set to 0, precompensation is inhibited. When set to 1 and with double-density encoding, write precompensation is applied to the following data/clock bit patterns:

Precomp Time	Data/Clock Pattern (In Data Shift Reg)			Bit Being Written	Bits Already Written to Disk		
2T (Late)	0	1	0	1	0	0	0
2T (Late)	0	1	0	1	0	0	1
2T (Early)	1	0	0	1	0	1	0
2T (Early)	0	0	0	1	0	1	0

where, $T = \frac{1}{\text{crystal frequency}}$ if bit 7 of CSR 2 ($^1/_2$F) = 1

$T = \frac{2}{\text{crystal frequency}}$ if bit 7 of CSR 2 ($^1/_2$F) = 0

Bit 1 (Read Mode)
When set to 0, the 8X330 reads data from the disk and transfers it to the Data Register; when set to 1, data from the Data Register is transferred to the disk, provided the Write Gate Enable bit (CSR1/Bit 0) is set to 1. With WGE set to 0 and the Read Mode bit set to 1, the current-controlled oscillator is forced to lock onto the crystal oscillator; this technique is used during a data-read operation to ensure rapid acquisition of the disk data.

Bits 2,3 (Bit Selects 1 and 0)
Together with the Sync Enable (CSR 1/Bit 3), these two bits allow the user to establish byte boundaries for the data stream; this is done in the following way. After bit synchronization is established, and the preamble pattern is verified, the 8X330 looks for a change in the normal preamble pattern. As shown in the following truth table, Bit Select 1 (Bit 2) and Bit Select 0 (Bit 3) identifies the bit cell within the first nibble of the first Address-Mark byte in which the first deviation from the normal preamble is expected. BYTRA is always referenced to bit cell 0.

BS 0	BS 1	Bit Cell
0	0	0
0	1	1
1	0	2
1	1	3

Bit 4 (Preamble Select)
This bit is used only for bit sychronization—refer to CSR 1/Bit 3. With Bit 1 of CSR 2 set to 0 (Read Mode) and the Preamble Select bit set to 0, the preamble field is assumed to be all zeroes; with the Preamble Select it set to 1, the preamble field is assumed to be all ones. In either case preamble validity is determined by the 8X300.

Bits 5,6 (E1 and E2)
Together, E1 and E2 select the encoding scheme used to write data on the disk—refer to truth table that follows

E1	E2	Encoding Scheme
1	X	FM
0	0	MFM
0	1	M²FM

x = don't care

Bit 7 (1/2F)
This bit allows the data transfer rate to be changed without modification of the frequency-selective components in the data-separation logic; thus, differences in data transfer rates between standard-and-mini floppies can be accommodated via software—no component or other hardware changes. Assuming an 8 MHz crystal and with the 1/2F bit set to 1, the data transfer rate is 250K-bits per second in the single-density (FM) mode and 500K-bits per second in the double-density (MFM/M²FM) mode. When set to 0, the transfer rates are halved—125K-bits and 250K-bits, respectively. When using frequencies other than 8 MHz, the data transfer rate is determined as follows:

Bit 7 ($^1/_2$F)	Single-Density (FM)	Double-Density (MFM/M²FM)
0	$\frac{\text{xtal freq}}{64}$	$\frac{\text{xtal freq}}{32}$
1	$\frac{\text{xtal freq}}{32}$	$\frac{\text{xtal freq}}{16}$

Command/Status Register #3 (CSR 3/Address 134_8)

This register contains seven bits (Bit 0 through Bit 6) which determines the state of the disk-command outputs; writing to Bit 7 has no effect and reading Bit 7 always returns a zero When a logical "1" is specified by the 8X300 program for a given disk-command line, a high will appear at the output of the 8X330 for that particular command line. Each bit and the output pin it controls are summarized below.

Bit (CSR 3)	Control Function	Pkg Pin No.
0	DC1 Output	12
1	DC2 Output	11
2	DC3 Output	10
3	DC4 Output	9
4	DC5 Output	8
5	DC6 Output	7
6	DC7 Output	6

Command/Status Register #4 (CSR 4/Address 135_8)

This register contains four bits (Bit 0 through Bit 3) which reflect the state of the disk-status inputs to the 8X330, reading all other bits (4 through 7) always returns a zero. These read-only bits and the reflected status they represent are as follows; the information specified by notation for Bit 7/CSR 1 is applicable to these input lines.

Bit (CSR 4)	Control Function	Pkg Pin No.
0	DS2 Input	16
1	DS3 Input	15
2	DS4 Input	14
3	DS5 Input	13

Sector Length Register—Address 136_8

This register contains the load value for the lower eight (LSBs) bits of the Byte Counter. Data is transferred from the Sector Length Register to the Byte Counter under control of Load Counter Bit in CSR 1. When the contents of this register are transferred to another location via a read or write commands, the original holding of data is not lost, thus, if the same data is to be used more than once, a repetitive read or write can be implemented without reloading the register.

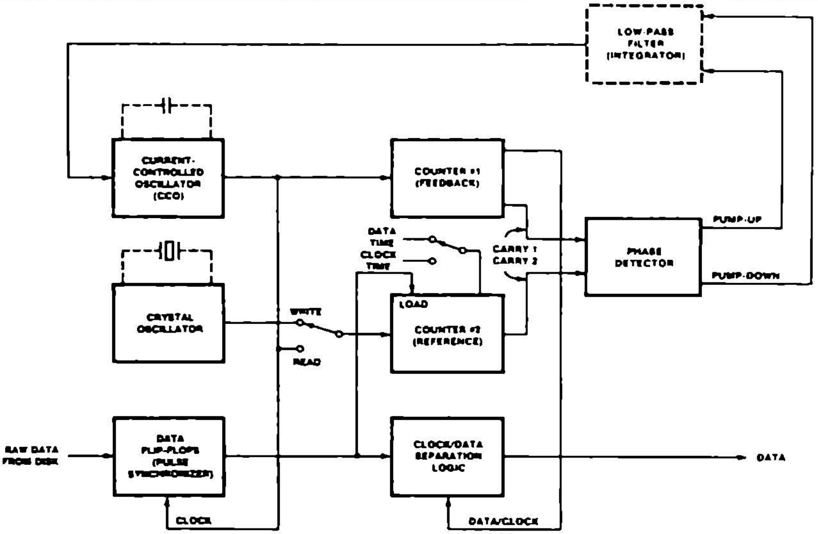
Data Register—Address 137_8

Together with the Data Shift Register, the Data Register is used for bidirectional transfer of data between the 8X330 and the I/O bus. All transfers to-and-from this register are made in conjunction with Bit 6 (BYTRA—Byte Transfer Flag) of CSR 1. When the Data Register Control bit (CSR 1/Bit 2) is set to 0, the content of this register is interleaved with four bits of data and four bits of clock. When data is transferred from the Data Register to the Data Shift Register, the original content of the Data Register is not lost.

Phase Lock Loop (PLL) and Data Separation Logic

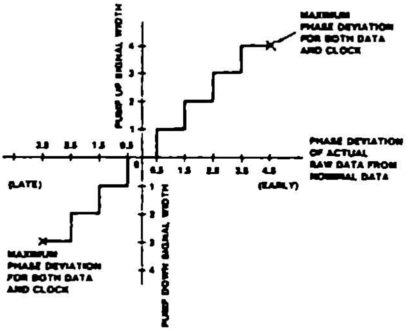
An expanded view of the phase-lock loop and the data-separation logic is shown in Figure 4. Basically, the PLL consists of two counters, a phase detector, and a feedback loop containing a low-pass filter (off-chip) that controls a phase-locked oscillator (CCO). In simplified form, the data-separation logic consists of data flip-flops (pulse synchronizer) and other circuits required to separate data and clock transitions. In the read mode, the output of the phase-locked oscillator (CCO) is applied to the clock inputs of counter #1, counter #2, and the pulse synchronization circuits. Essentially, the frequencies of the two counters are identical (phase relationships may or may not be identical); to maintain proper frequencies and to continuously correct for any phase deviations, the following actions occur.

Preset values which represent, respectively, nominal mid-points of the clock and data windows are present at counter #2 and, when an output appears at the pulse synchronizer, these preset values are entered. The count sequence for both counters is from "0 to F", hence, the phase difference between Carry 1 (counter #1) and Carry 2 (counter #2) actually corresponds to any phase deviation between the CCO and the synchronized data from the disk. The phase detector measures the phase difference between the two carry inputs and produces a series of quantized pulses whose widths are proportional to the phase error at the end of each counting cycle. After integration by the low-pass filter, a current proportional to the phase error is applied to the current-controlled oscillator. Accordingly, the CCO is driven in a direction (pump-up or pump-down) to correct any phase difference between the synchronized disk data and the feedback-controlled clock. Phase detector characteristics for both single-and-double density formats are shown in Figures 5 and 6.


LOW-PASS FILTER (INTEGRATOR)
CURRENT-CONTROLLED OSCILLATOR (CCO)
COUNTER #1 (FEEDBACK)
DATA TIME
CLOCK TIME
CARRY 1
CARRY 2
PHASE DETECTOR
PUMP-UP
PUMP-DOWN
CRYSTAL OSCILLATOR
WRITE
READ
LOAD
COUNTER #2 (REFERENCE)
RAW DATA FROM DISK
DATA FLIP-FLOPS (PULSE SYNCHRONIZER)
CLOCK/DATA SEPARATION LOGIC
DATA
CLOCK
DATA/CLOCK WINDOWS

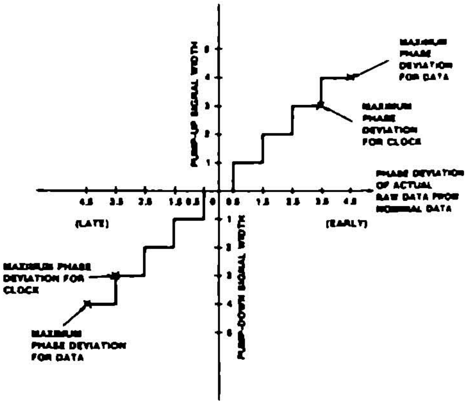
NOTE
Components shown with dashed lines are located off-chip

Figure 4. Simplified Block of Phase-Lock Loop



NOTE
All horizontal/vertical units are specified in CCO clock periods; for Standard Floppy FM Format, one CCO clock period equals 250 nanoseconds

Figure 5. Phase Detector Characteristic for Single-Density (FM) Format



NOTE
All horizontal/vertical units are specified in CCO clock periods; for standard floppy double density (MFM/M²FM), one CCO clock period equals 125 nanoseconds

Figure 6. Phase Detector Characteristic for Double-Density (MFM/M²FM) Format

VALVO

Data Processing and Error-Check Functions

These functions of the 8X330 are summarized in Figures 7 and 8. The read/write operations are software-controlled by previously-described bits of command/status registers CSR1 and CSR2. For the sake of simplicity, control lines and much of the control logic associated with the data processing and error-check functions are omitted in the read/write diagrams.

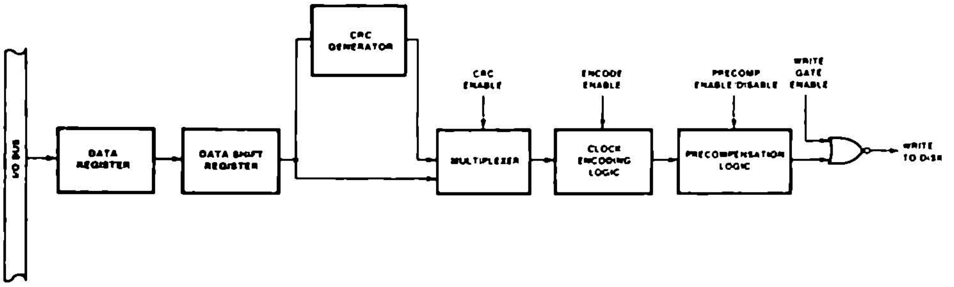

I/O BUS
DATA REGISTER
DATA SHIFT REGISTER
CRC GENERATOR
CRC ENABLE
MULTIPLEXER
ENCODE ENABLE
CLOCK ENCODING LOGIC
PRECOMP ENABLE DISABLE
PRECOMPENSATION LOGIC
WRITE GATE ENABLE
WRITE TO DISK

Figure 7. Simplified Block of Data Processing and Error Check Functions—Write Mode

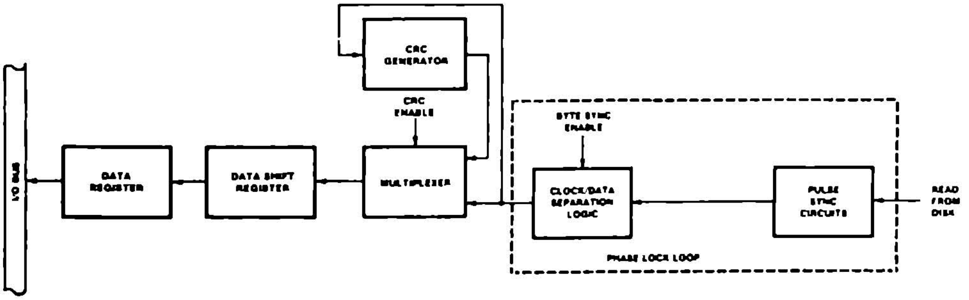

I/O BUS
DATA REGISTER
DATA SHIFT REGISTER
CRC GENERATOR
CRC ENABLE
MULTIPLEXER
BYTE SYNC ENABLE
CLOCK/DATA SEPARATION LOGIC
PULSE SYNC CIRCUITS
PHASE LOCK LOOP
READ FROM DISK

Figure 8. Simplified Block of Data Processing and Error Check Functions—Read Mode

DC CHARACTERISTICS V_{CC} = 5V (±5%), T_A = 0°C to 70°C

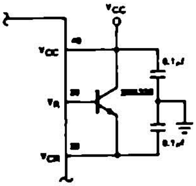
PARAMETER		TEST CONDITIONS	LIMITS			UNITS	COMMENTS
			Min	Typ	Max		
V_{IH}	High level input voltage		2.0		V_{CC}	V	For all inputs except X1, X2, C1, C2, CCO, and V_{CR}
V_{IL}	Low level input voltage		−1.0		.8	V	
V_{CC}	Supply voltage		4.75	5.0	5.25	V	5V (±5%)
V_{CR}	Regulator voltage	V_{CC} = 5V		3.1		V	From series-pass transistor
V_{CL}	Input clamp voltage	V_{CC} = Min I_{IN} = −5mA	−1.0			V	Inputs X1, X2, C1, C2, and CCO do not have internal clamp diodes.
V_{OH}	High level output voltage	V_{CC} = Min; I_{OH} = −0.4mA	2.7			V	DC1 through DC7 (Pins 6-12) & $\overline{DW}$ (Pin 4)
		V_{CC} = Min; I_{OH} = −3mA	2.4			V	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
V_{OL}	Low level output voltage	V_{CC} = Min; I_{OL} = 8mA			0.5	V	DC1 through DC7 (Pins 6-12), $\overline{PUP}$, $\overline{PDN}$ (Pins 33 34), $\overline{DW}$ (Pin 4)
		V_{CC} = Min; I_{OL} = 16mA			0.55	V	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
		V_{CC} = Min; I_{OL} = 40mA			0.55	V	$\overline{WG}$ (Pin 19)
I_{CEX}	Open-collector leakage current with output set to 1.	V_{CC} = Min; V_{OUT} = V_{CC}			100	µA	$\overline{WG}$ (Pin 19), $\overline{PUP}$ (Pin 34), $\overline{PDN}$ (Pin 33)
I_{IH}	High level input current	V_{CC} = Max; V_{IN} = 2.7V			20	µA	DS1-DS5 (Pins 13-17), $\overline{PF}$ (Pin 18); $\overline{DR}$ (Pin 5)
					40	µA	$\overline{ME}$ (Pin 21); MCLK (Pin 22), SC (Pin 23), WC (Pin 24)
		V_{CC} = Max; V_{IN} = 5.25V; CCO (Pin 35) input current = 0mA			4	mA	With C1 (Pin 36) under test, C2 (Pin 37) is open and, vice-versa
		V_{CC} = Max; V_{IN} = 5.25V CCO (Pin 35) input current = 1mA			2	mA	
		V_{CC} = Max; V_{IN} = 0.6V			4	mA	With X1 (Pin 2) under test, X2 (Pin 3) is open and, vice-versa
		V_{CC} = Max; V_{IN} = 4.5V			50	µA	$\overline{IV0}$ - $\overline{IV7}$ (pins 25-32)
V_{CCO}	Input voltage for current-controlled oscillator	V_{CC} = 5V; T_A = 25°C CCO input current (Pin 35) = 300 µA		750		mV	

DC CHARACTERISTICS (Cont'd) V_{CC} = 5V (±5%), T_A = 0°C to 70°C

PARAMETER		TEST CONDITIONS	LIMITS			UNITS	COMMENTS
			Min	Typ	Max		
I_{IL}	Low level input current	V_{CC} = Max; V_{IN} = 0.4V			−400	µA	DS1-DS5 (Pins 13-17); $\overline{PF}$ (Pin 18); $\overline{DR}$ (Pin 5)
					−800	µA	$\overline{ME}$ (Pin 21); MCLK (pin 22); SC (Pin 23); WC (Pin 24)
					−4	mA	X1 (Pin 2), X2 (Pin 3) — with X1 under test, X2 is open and, vice-versa; (Note 4)
		V_{CC} = Max V_{IN} = 0.5V			−550	µA	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
I_{OS}	Output short-circuit current	V_{CC} = Max; Output = "1"; V_{OUT} = "0".	−15		−100	mA	DC1-DC7 (Pins 6-12) & $\overline{DW}$ (Pin 4)
		(NOTE At any time, no more than one output should be connected to ground)	−30		−140	mA	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
I_{CC}	(Pin 40)	V_{CC} = Max			200	mA	
I_{CR}		V_{CC} = Max			260	mA	
I_{REG}	(Pin 39)	V_{CC} = 5V; V_{CR} = 0V & V_R = 2V	−16		−24	mA	

NOTES

1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.
3. Unless otherwise specified, each test requires that V_{CR} be supplied through a series-pass transistor as shown in the accompanying drawing.
4. The recommended interfacing technique for C_1 / C_2 is discussed on page 6-73.



AC CHARACTERISTICS $V_{CC} = 5V (\pm 5\%)$, $T_A = 0°C$ to 70°C

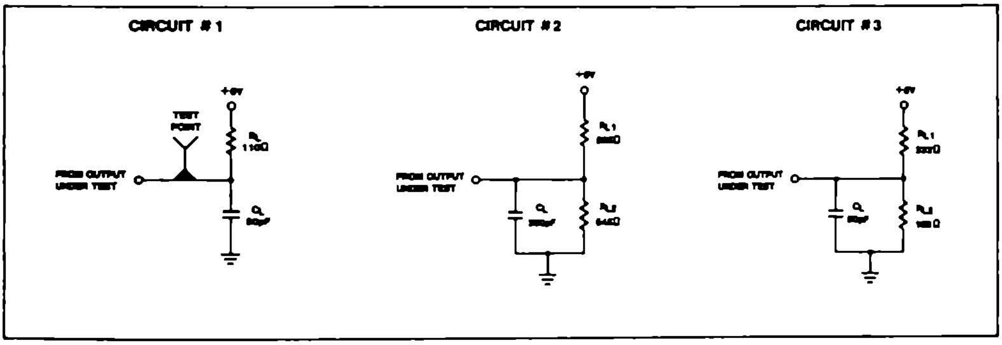
MNEMONIC	REFERENCE	INPUT	OUTPUT	Min	Typ	Max	COMMENTS
t_{PD}		╲ $\overline{PF}$	$\overline{WG}$ ╱			30ns	Refer to Note 3 and Test Loading Circuit #1
t_{PD}		╱ MCLK	$\overline{WG}$ ╱			70ns	
t_{PD}		╱ MCLK	$\overline{WG}$ ╲			70ns	
t_{PD}		╱ MCLK	DC1-7 ╱			70ns	Refer to Note 3 and Test Loading Circuit #2.
t_{PD}		╱ MCLK	DC1-7 ╲			70ns	
t_{pw}		╲ $\overline{PF}$ ╱		50ns			
t_{pw}		╲ $\overline{DR}$ ╱		50ns			
t_{pw}			╲ $\overline{DW}$ ╱				Note 1
t_{SETUP}	╲ SC	Input on DS1-5		55ns			Note 2
t_{SETUP}	╲ WC	Input on DS1-5		55ns			Note 2
t_{SETUP}	╲ $\overline{ME}$	Input on DS1-5		55ns			Note 2
t_{HOLD}	╲ SC	Input on DS1-5		0ns			Note 2
t_{HOLD}	╲ WC	Input on DS1-5		0ns			Note 2
t_{HOLD}	╲ $\overline{ME}$	Input on DS1-5		0ns			Note 2
t_{OE}—$\overline{ME}$, SC & WC		╲ $\overline{ME}$ ╲ SC ╲ WC	I/O bus			25ηsec	Refer to Test Loading Circuit #3
t_{OD}—$\overline{ME}$, SC & WC		$\overline{ME}$ ╱ SC ╱ WC ╱	I/O bus (three-state)			30ηsec	
t_w (MCLK pulse width)		╱ MCLK ╲		45ns			
t_{SD} (data setup time)	MCLK ╲	I/O bus		50ns			
t_{SD} ($\overline{ME}$ setup time)	MCLK ╲	$\overline{ME}$		45ns			
t_{SD} (SC setup time)	MCLK ╲	SC		45ns			
t_{SD} (WC setup time)	MCLK ╲	WC		45ns			
t_{HD} (data hold time)	MCLK ╲	I/O bus		0ns			

AC CHARACTERISTICS (Cont'd) V_{CC} = 5V (±5%), T_A = 0°C to 70°C

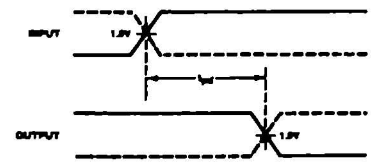
MNEMONIC	REFERENCE	INPUT	OUTPUT	Min	Typ	Max	COMMENTS
t_{HD} ($\overline{ME}$ hold time)	MCLK ↘	$\overline{ME}$		0ns			
t_{HD} (SC hold time)	MCLK ↘	SC		0ns			
t_{HD} (WC hold time)	MCLK ↘	WC		0ns			

NOTES
1. Write pulse width = 2/F_{XTAL}, that is, for 8MHz crystal, t_{pw} = 250usec (typical)
2. Changes on DB1-8 are not stored in read mode ($\overline{ME}$ = 0, SC = 0, and WC = 0)
3. During the period when MCLK is high, measurement is made with $\overline{ME}$ = Low, SC = Low, and WC = High.

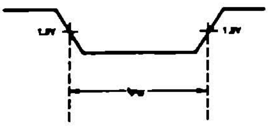
TEST LOADING CIRCUIT


CIRCUIT #1
CIRCUIT #2
CIRCUIT #3
+5V
TEST POINT
FROM OUTPUT UNDER TEST
R_L
C_L
R_{L1}
R_{L2}

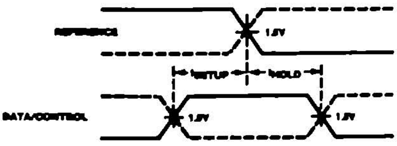
PROPAGATION DELAY TIME


INPUT
1.5V
t_{pd}
OUTPUT
1.5V

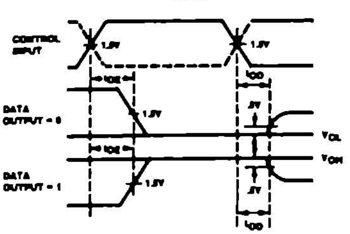
PULSE WIDTH WAVEFORM



SETUP & HOLD TIMES


REFERENCE
1.5V
t_{SETUP}
t_{HOLD}
DATA/CONTROL
1.5V
1.5V

OUTPUT ENABLE AND DISABLE TIMES


CONTROL INPUT
1.5V
1.5V
t_{OE}
t_{OD}
DATA OUTPUT = 0
1.5V
.5V
V_{OL}
V_{OH}
t_{OE}
DATA OUTPUT = 1
1.5V
.5V
t_{OD}

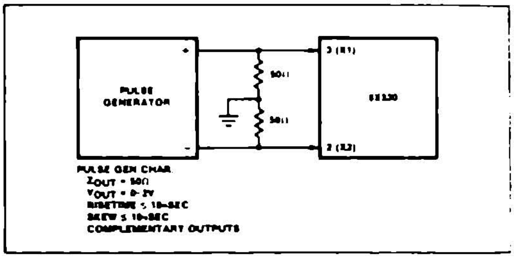
CLOCK REQUIREMENTS

Crystal Oscillator. The on-chip crystal oscillator circuit is designed for operation using an external series-resonant quartz crystal; alternately the crystal oscillator can be driven with complementary outputs of a pulse generator or interfaced to a master clock source via TTL logic—see accompanying circuits. When a crystal is used, the on-chip oscillator operates at the resonant frequency (f.) of the crystal; the crystal connects to the 8X330 via pins 3 (X1) and 2 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, avoid close proximity to all potential noise sources. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

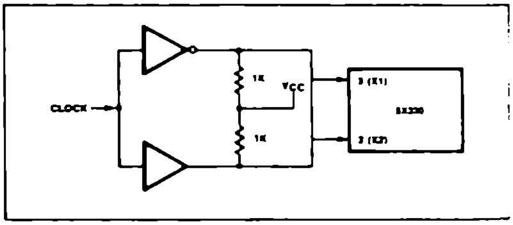
Type: Fundamental mode, series resonant
Impedance at Fundamental: 35-ohms maximum
Impedance at Harmonics and Spurs: 50-ohms minimum

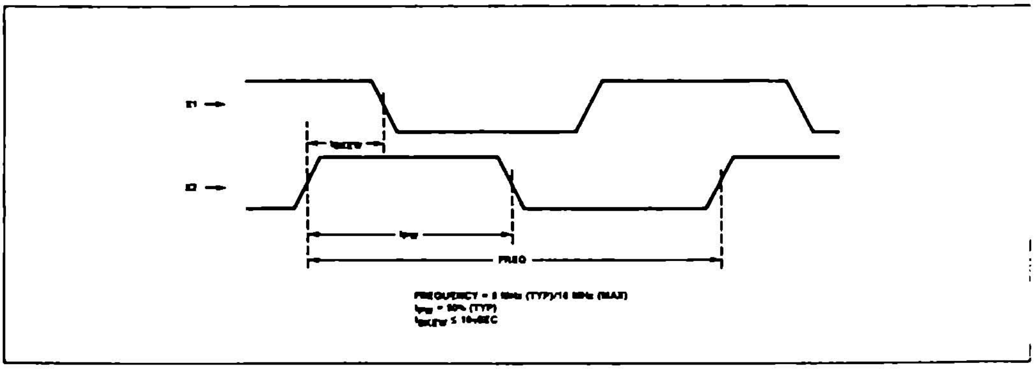
When the crystal oscillator is externally-driven, typical waveforms are as follows:

CLOCKING XTAL OSC WITH PULSE GEN


PULSE GENERATOR
+
−
50Ω
50Ω
3 (X1)
8X330
2 (X2)
PULSE GEN CHAR.
ZOUT = 50Ω
VOUT = 0-3V
RISETIME ≤ 10nSEC
SKEW ≤ 10nSEC
COMPLEMENTARY OUTPUTS

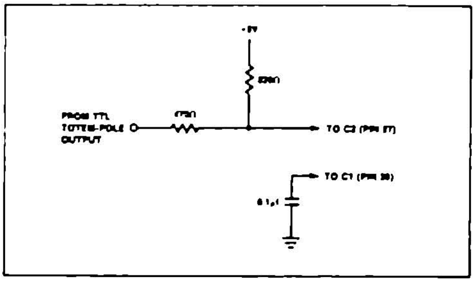
CLOCKING XTAL OSC WITH OPEN-COLLECTOR TTL


CLOCK
1K
VCC
1K
3 (X1)
8X330
2 (X2)

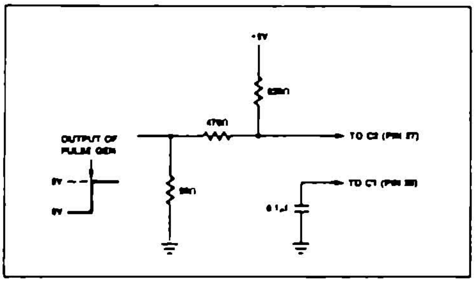

X1
X2
tSKEW
tPW
FREQ
FREQUENCY = 8 MHz (TYP)/16 MHz (MAX)
tPW = 50% (TYP)
tSKEW ≤ 10nSEC

Current-Controlled Oscillator (CCO).
A non-polarized ceramic or mica capacitor is recommended for the current-controlled oscillator. The capacitor connects to the 8X330 via pins 37 (C2) and 36 (C1); lead lengths of the capacitor should be approximately the same and as short as possible. When the input current to the CCO is near zero (maximum frequency), the capacitor value should be chosen so that the high-limit rest frequency of the oscillator does not exceed 24 MHz. If the rest frequency is higher than 24 MHz, synchronization of the CCO with the crystal oscillator just prior to the read operation, may be impeded. The curves in Figure 9 (current-versus-frequency) and Figure 8 (capacitance-versus-frequency) show how these design parameters affect operation of the CCO over a temperature range of 0° C to 70° C. A suitable test circuit for verification/validation of the current-controlled oscillator is also shown in Figure 10. Like the crystal oscillator, the CCO can be driven with the TTL output of a pulse generator or interfaced to a master clock via TTL logic—see accompanying diagrams.

CLOCKING WITH OPEN-COLLECTOR TTL



CLOCKING WITH PULSE GENERATOR



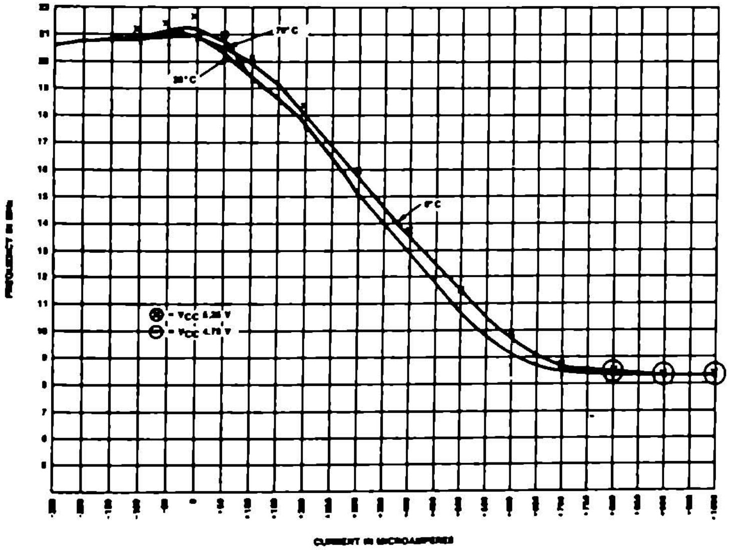


Figure 9. Current-versus-Frequency with: V_{CC} = 5V and Capacitance = 25 Picofarads

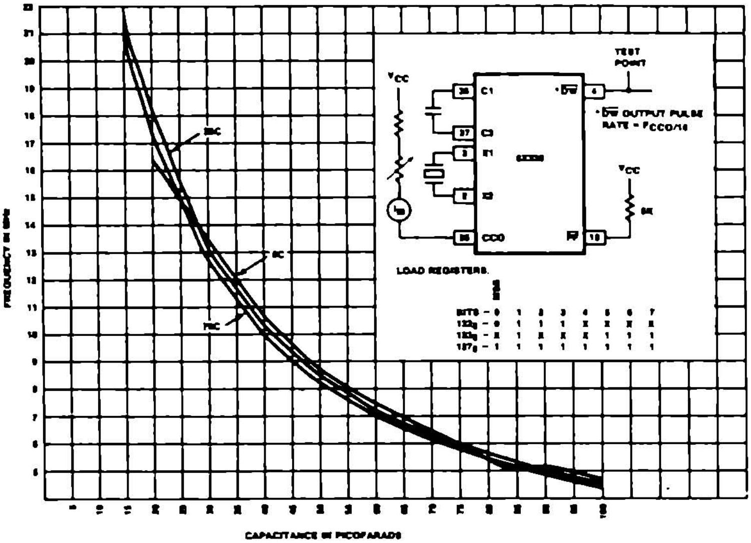
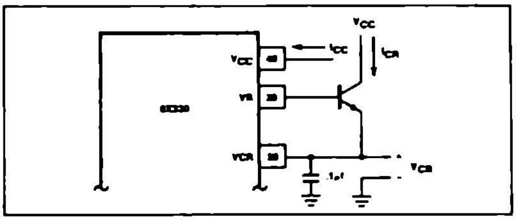


Figure 10. Capacitance-versus-Frequency with: V_{CC} = 5V, VCR = 2.5V, and I = 300μA

VOLTAGE REGULATOR

All internal logic of the 8X330 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in accompanying diagrams. To minimize lead inductance, the transistor should be as close as possible to the 8X330 package and the emitter should be ac-grounded via a 0.1-microfarad capacitor.

TYPICAL HOOK-UP



ELECTRICAL SPECIFICATIONS

*PARAMETER	CONDITIONS	LIMITS
h_{fe}	V_{CE} = 2V	>50
V_{BEON}	V_{CE} = 5V/I_C = 500mA	<1V
V_{CESAT}	I_C = 500 mA/I_B = 50 mA	<0 5V
BV_{CEO}		>8V
f_t		>30 MHz

*Medium power NPN silicon (0°<T_A<70°C) recommended parts 2N5320 2N5

ORDERING INFORMATION

Order number: N8X330 N

Supply voltage: 5V (±5%)
Operating temperature range: 0°C to +70°C

PRELIMINARY SPECIFICATION

DESCRIPTION

The 8X350 bipolar RAM is designed principally as a working storage element in an 8X300 based system. Internal circuitry is provided for direct use in 8X300 applications. When used with the 8X300, the RAM address and data buses are tied together and connected to the IV bus of the system.

The data inputs and outputs share a common I/O bus with 3-state outputs. The address inputs can be operated in a transparent or latched mode.

The 8X350 is available in commercial and military temperature ranges. For the commercial temperature range (0°C to +75°C) specify N8X350-F, and for the military temperature range (−55°C to +125°C) specify S8X350-F.

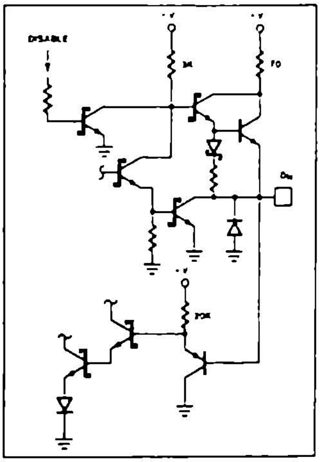
FEATURES

- On-chip address latches
- 3-state outputs
- Schottky clamped TTL
- Internal control logic for 8X300 system
- Directly interfaces with the 8X300 bipolar microprocessor with no external logic

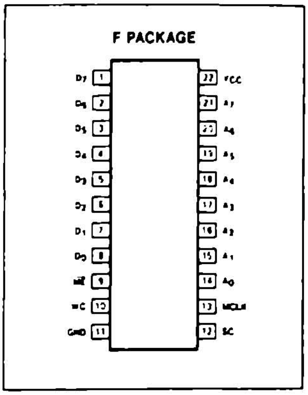
APPLICATIONS

- 8X300 working storage

TYPICAL I/O STRUCTURE


DISABLE
+V
1K
+V
FO
D_n
+V
20K

PIN CONFIGURATION


F PACKAGE
D7 1
D6 2
D5 3
D4 4
D3 5
D2 6
D1 7
D0 8
ME 9
WC 10
GND 11
22 VCC
21 A7
20 A6
19 A5
18 A4
17 A3
16 A2
15 A1
14 A0
13 MCLK
12 SC

ABSOLUTE MAXIMUM RATINGS

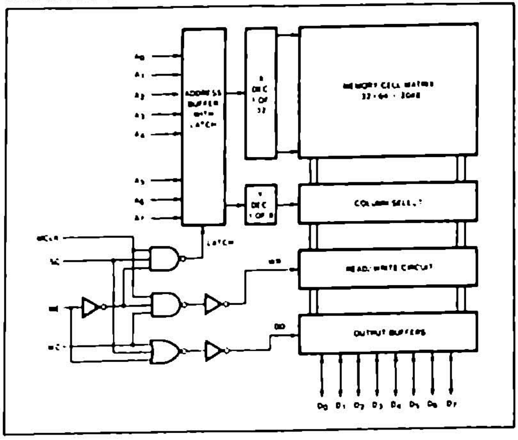
	PARAMETER[1]	RATING	UNIT
V_{CC}	Supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
	Output voltage		Vdc
V_{OH}	High	+5.5	
V_O	Off-state	+5.5	
	Temperature range		°C
T_A	Operating		
	N8X350	0 to +75	
	S8X350	−55 to +125	
T_{STG}	Storage	−65 to +150	

TRUTH TABLE

Note X = Don't care

MODE	$\overline{ME}$	SC	WC	MCLK	BUSSED DATA/ADDRESS LINES
Hold address Disable data out	1	X	X	X	High Z data out
Input new address	0	1	0	1	Address
Hold address Disable data out	0	1	0	0	High Z data out
Hold address Write data	0	0	1	1	Data in
Hold address Disable data out	0	0	1	0	High Z data out
Hold address Read data	0	0	0	X	Data out
Undefined state[13]	0	1	1	1	—
Hold address[13] Disable data out	0	1	1	0	High Z data out

BLOCK DIAGRAM


A0
A1
A2
A3
A4
A5
A6
A7
ADDRESS BUFFER WITH LATCH
X DEC 1 OF 32
MEMORY CELL MATRIX 32 × 64 = 2048
Y DEC 1 OF 8
COLUMN SELECT
READ/WRITE CIRCUIT
OUTPUT BUFFERS
LATCH
MCLK
SC
ME
WC
WE
OD
D0 D1 D2 D3 D4 D5 D6 D7

DC ELECTRICAL CHARACTERISTICS[2]

N8X350: $0^\circ C \le T_A \le +75^\circ C$, $4.75V \le V_{CC} \le 5.25V$
S8X350: $-55^\circ C \le T_A \le +125^\circ C$, $4.5V \le V_{CC} \le 5.5V$

PARAMETER		TEST CONDITIONS	N8X350			S8X350			UNIT
			Min	Typ[3]	Max	Min	Typ[3]	Max	
	Input voltage								V
V_{IL}	Low[1]	V_{CC} = Min			.85			.80	
V_{IH}	High[1]	V_{CC} = Max	2.0			2.0			
V_{IC}	Clamp[1,4]	V_{CC} = Min, I_{IN} = −12mA		−0.8	−1.2		−.8	−1.2	
	Output voltage	V_{CC} = Min							V
V_{OL}	Low[1,5]	I_{OL} = 9.6mA		0.35	0.5			.5	
V_{OH}	High[1,6]	I_{OH} = −2mA	2.4	3.3		2.4			
	Input current								μA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100			−150	
I_{IH}	High	V_{IN} = 5.5V			25			50	
	Output current								
$I_{O(OFF)}$	High Z state	V_{OUT} = 5.5V			40			60	μA
		V_{OUT} = 0.5 V[7]			−100			−100	μA
I_{OS}	Short circuit[4,7]	V_{OUT} = 0V	−20		−70	−15		−85	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = Max		135	185			185	mA
	Capacitance	V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V $\overline{ME}$ = V_{IH}		8			8		

AC ELECTRICAL CHARACTERISTICS[2,10]

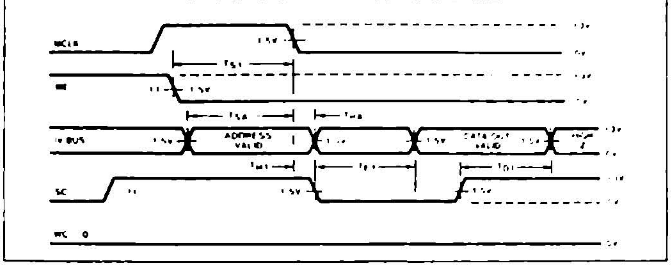
N8X350: $0^\circ C \le T_A \le +75^\circ C$, $4.75V \le V_{CC} \le 5.25V$
S8X350: $-55^\circ C \le T_A \le +125^\circ C$, $4.5V \le V_{CC} \le 5.5V$

PARAMETER		TO	FROM	N8X350			S8X350			UNIT
				Min	Typ[3]	Max	Min	Typ[3]	Max	
	Enable time									ns
T_{E1}	Output	Data out	SC−			35			40	
T_{E2}	Output	Data out	$\overline{ME}$−			35			40	
	Disable time									ns
T_{D1}	Output	Data out	SC+			35			40	
T_{D2}	Output	Data out	$\overline{ME}$+			35			40	
	Pulse width									ns
T_W	Master clock[9]			40			50			
	Setup and hold time									ns
T_{SA}	Setup time	MCLK−	Address	30			40			
T_{HA}	Hold time	Address	MCLK−	5			10			
T_{SD}	Setup time	MCLK−	Data in	35			45			
T_{HD}	Hold time	Data in	MCLK−	5			10			
T_{S3}	Setup time	MCLK−	$\overline{ME}$−	40			50			
T_{H3}	Hold time	$\overline{ME}$+	MCLK−	5			5			
T_{S1}	Setup time	MCLK−	$\overline{ME}$−	30			40			
T_{H2}	Hold time	$\overline{ME}$−	MCLK−	5			5			
T_{S2}	Setup time	$\overline{ME}$−	SC−, WC−	0			5			
T_{H1}	Hold time	SC−	MCLK−	5			5			
T_{H4}	Hold time	WC−	MCLK−	5			5			

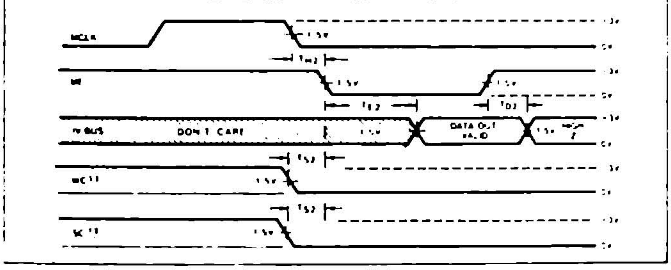
PRELIMINARY SPECIFICATION

TIMING DIAGRAMS

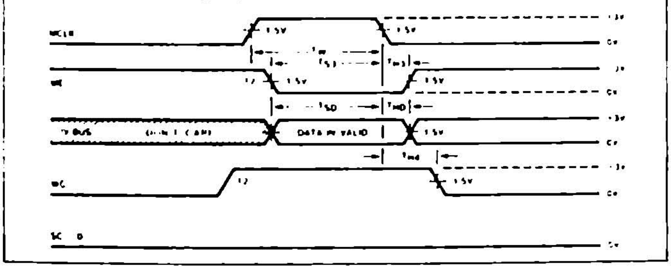
TRANSPARENT ADDRESS READ CYCLE



LATCHED ADDRESS READ CYCLE



LATCHED ADDRESS WRITE CYCLE



NOTES

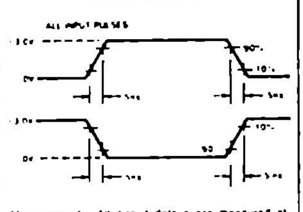
1. All voltage values are with respect to network ground terminal.
2. The operating ambient temperature ranges are guaranteed with transverse air flow exceeding 400 linear feet per minute and a 2 minute warm up.
   Typical thermal resistance values of the package at maximum temperature are:
   θ_{JA} junction to ambient at 400fpm air flow 50°C/watt
   θ_{JA} junction to ambient, still air 90°C/watt
   θ_{JA} junction to case 20°C/watt
3. All typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
4. Test each pin one at a time.
5. Measured with a logic low stored. Output sink current is supplied through a resistor to V_{CC}.
6. Measured with a logic high stored.
7. Duration of the short circuit should not exceed 1 second.
8. I_{CC} is measured with the write enable and memory enable inputs grounded, all other inputs at 4.5V and the output open.
9. Minimum required to guarantee a Write into the slowest bit.
10. Applied to the 8X300 based system with the data and address pins tied to the IV Bus.
11. $SC + \overline{ME} = 1$ to avoid bus conflict.
12. $WC + \overline{ME} = 1$ to avoid bus conflict.
13. The SC and WC outputs from the 8X300 are never at 1 simultaneously.

TIMING DEFINITIONS

T_{S1}	Required delay between beginning of Master Enable low and falling edge of Master Clock.
T_{SA}	Required delay between beginning of valid address and falling edge of Master Clock.
T_{HA}	Required delay between falling edge of Master Clock and end of valid Address.
T_{H1}	Required delay between falling edge of Master Clock and when Select Command becomes low.
T_{E1}	Delay between beginning of Select Command low and beginning of valid data output on the IV Bus.
T_{D1}	Delay between when select Command becomes high and end of valid data output on the IV Bus.
T_{H2}	Required delay between falling edge of Master Clock and when Master Enable becomes low.
T_{E2}	Delay between when Master Enable becomes low and beginning of valid data output on the IV Bus.
T_{D2}	Delay between when Master Enable becomes high and end of valid data output on the IV Bus.
T_{S2}	Required delay between when Select Command or Write Command becomes low and when Master Enable becomes low.
T_W	Minimum width of the Master Clock pulse.
T_{S3}	Required delay between when Master Enable becomes low and falling edge of Master Clock.
T_{H3}	Required delay between falling edge of Master Clock and when Master Enable becomes high.
T_{SD}	Required delay between beginning of valid data input on the IV Bus and falling edge of Master Clock.
T_{HD}	Required delay between falling edge of Master Clock and end of valid data input on the IV Bus.
T_{H4}	Required delay between falling edge of Master Clock and when Write Command becomes low.

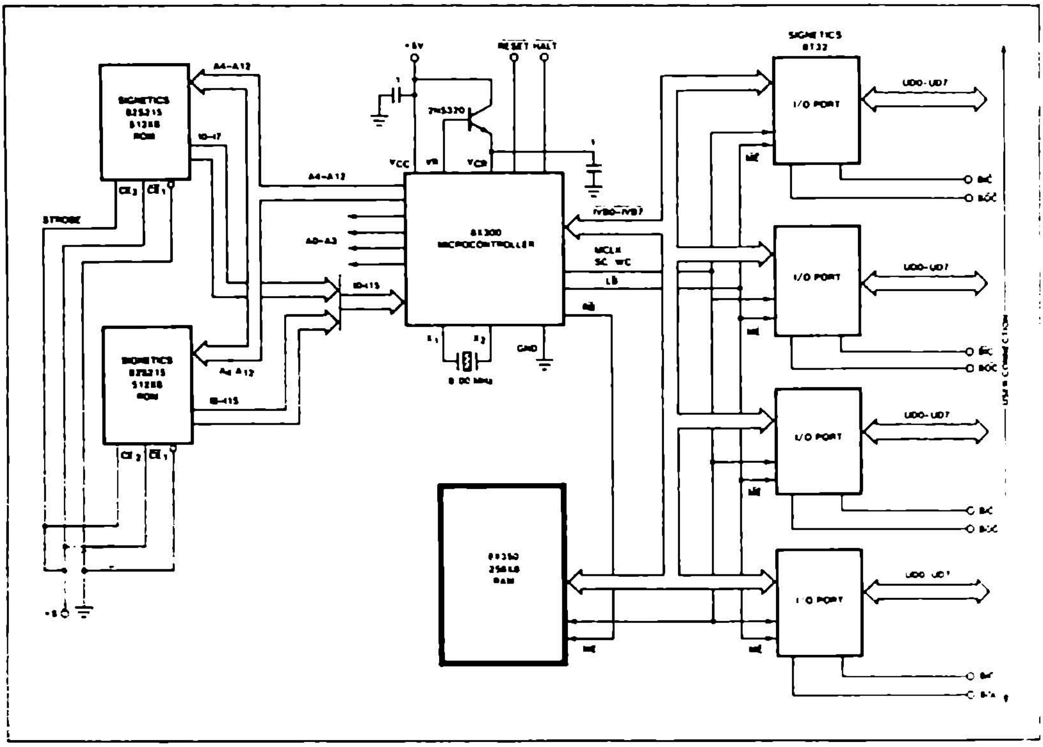
VOLTAGE WAVEFORM

INPUT PULSES



Measurements: All circuit delays are measured at the +1.5V level of inputs and output.

TYPICAL 8X350 APPLICATION


SIGNETICS
82S215
512X8
ROM
A4–A12
I0–I7
CE2
CE1
STROBE
SIGNETICS
82S215
512X8
ROM
A4 A12
I8–I15
CE2
CE1
+5
+5V
RESET
HALT
74S320
VCC
VR
VCR
A4–A12
A0–A3
I0–I15
8X300
MICROCONTROLLER
X1
X2
8.00 MHz
GND
IVB0–IVB7
MCLK
SC WC
LB
RB
8X350
256X8
RAM
ME
SIGNETICS
8T32
I/O PORT
UD0-UD7
ME
BIC
BOC
I/O PORT
UD0-UD7
ME
BIC
BOC
I/O PORT
UD0-UD7
ME
BIC
BOC
I/O PORT
UD0 UD7
ME
BIC
BOC
USER CONNECTION

JANUARY 1982

32 BYTE BIPOLAR RAM (32×8)

8X353

FEATURES

- ON-CHIP ADDRESS LATCHES
- THREE-STATE OUTPUTS
- DIRECTLY COMPATIBLE WITH THE 8X305 MICROCONTROLLER
- BIPOLAR ECL AND LOW-POWER SCHOTTKY TECHNOLOGY
- SINGLE +5 VOLT POWER SUPPLY
- 20-PIN DUAL IN-LINE PACKAGE

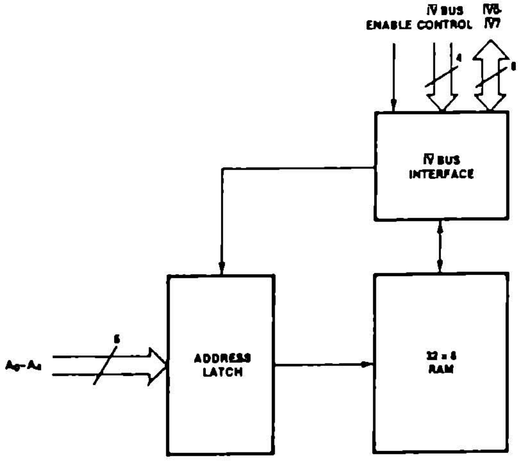
PRODUCT DESCRIPTION

The 8X353 32 Byte Bipolar RAM provides increased working storage capacity to 8X305 based systems. Its function is similar to that of the 8X350, except that it has less storage capacity and does not require a full bank of 8X305 addresses.

The 8X353 is organized as 32 8-bit words of memory. It is designed to directly connect to the Interface Vector ($\overline{\text{IV}}$) Bus of the 8X305 MicroController with no requirement for additional circuitry. The $\overline{\text{IV}}$ Bus is used to transmit both address and data to and from the device.

Addressing of the 8X353 can be accomplished using standard 8X305 I/O addressing techniques. In systems where extremely high performance is required, "extended microcode" techniques often used with the 8X300 Family can be readily employed.

8X353 32 BYTE RAM


$\overline{\text{IV}}$ BUS
ENABLE
CONTROL
$\overline{\text{IV}}$0-
$\overline{\text{IV}}$7
4
8
$\overline{\text{IV}}$ BUS
INTERFACE
A_0-A_4
5
ADDRESS
LATCH
32 × 8
RAM

JANUARY 1982

32 BYTE LIFO MEMORY (32×8) 8X355

FEATURES

- LAST-IN-FIRST-OUT STACK ORGANIZATION
- THREE-STATE OUTPUTS
- DIRECTLY COMPATIBLE WITH THE 8X305 MICROCONTROLLER
- BIPOLAR ECL AND LOW-POWER SCHOTTKY TECHNOLOGY
- SINGLE +5 VOLT POWER SUPPLY
- 20-PIN DUAL IN-LINE PACKAGE

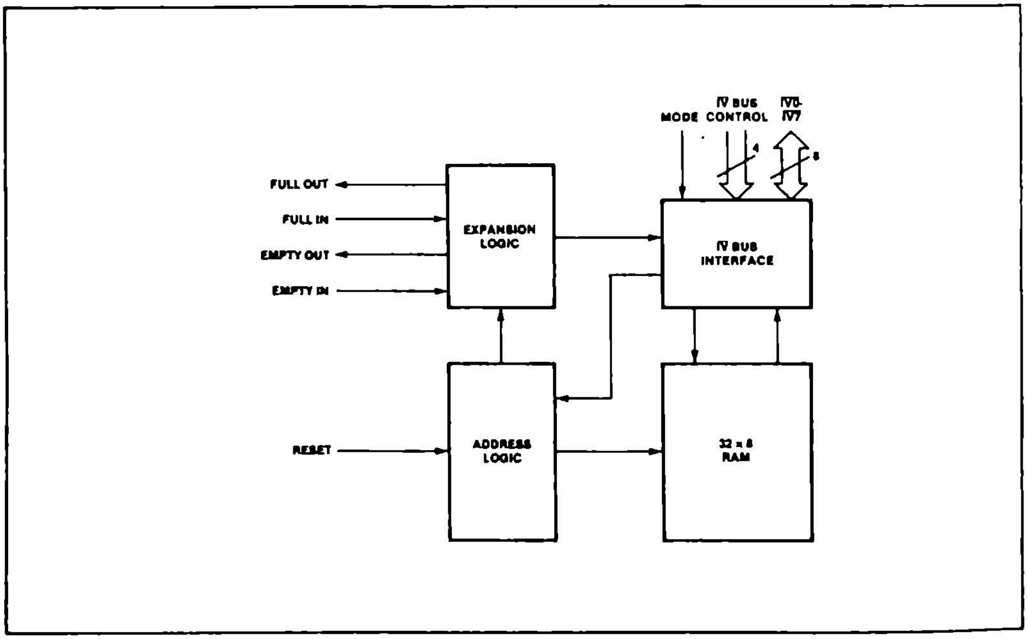
PRODUCT DESCRIPTION

The 8X355 32 Byte LIFO Memory provides an easy to use push-down stack capability to the 8X305 MicroController. Examples of its use are storing processor state data from the 8X305 during interrupt processing or operand data for mathematical operations. The LIFO stack organization eliminates the need to transmit an address for each data transfer. This results in optimum memory system efficiency when random access is not required.

The 8X355 contains 32 8-bit words of memory. This can be expanded by adding 8X355s without using additional MicroController address space simply by cascading the device. Pins are included to facilitate this function.

Addressing of the 8X355 can be accomplished using standard 8X305 I/O addressing techniques. In systems where extremely high performance is required, "extended microcode" techniques often used with the 8X300 Family can be readily employed.

8X355 LIFO BLOCK DIAGRAM



MEMORY ADDRESS DIRECTOR

8X360

FEATURES

- AUTOMATIC MEMORY ADDRESS MANAGEMENT
- 16-BIT ADDRESS SUPPORT
- 11 CONTROL REGISTERS FOR MAXIMUM FLEXIBILITY
- DIRECTLY COMPATIBLE WITH THE 8X305 MICROCONTROLLER
- BIPOLAR ISL AND LOW-POWER SCHOTTKY TECHNOLOGY
- SINGLE +5 VOLT POWER SUPPLY
- 40-PIN DUAL IN-LINE PACKAGE

PRODUCT DESCRIPTION

The 8X360 Memory Address Director provides all of the circuitry required to enable the efficient use of the 8X300 MicroController Family in applications demanding large working storage and high speed data transfer. The system can be operated at significantly higher data rates than were previously possible since address management tasks are handled automatically and off-loaded from the processor and its software.

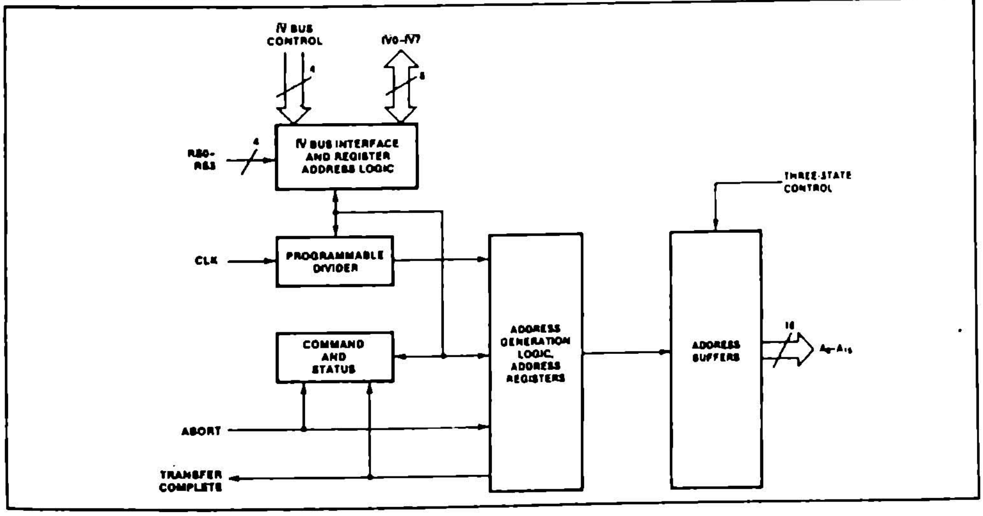
The 8X360 attaches directly to the Interface Vector ($\overline{IV}$) Bus of the 8X305 MicroController with no requirement for additional circuitry. All addresses contained in it are 16 bits in length, thus permitting attachment of up to 64K words of working storage through a single 8X360 device.

Address registers in the 8X360 are automatically incremented or decremented at the users option. The address updating can be controlled either from an external clock or through recognition of a specific port address on the 8X305 $\overline{IV}$ Bus. The clock can be prescaled from 1 to 64 to facilitate various applications and speeds.

The 8X360 is designed to make its use as simple as possible and require minimum overhead from the MicroController. To accomplish a memory transfer operation, the 8X305 supplies the Memory Address Director with a starting address and either an ending address or a block length. These addresses and/or lengths are preserved in a second set of internal registers in the 8X360 to facilitate efficient system operation during error handling or reinitialization. As the operation proceeds, status information is maintained in an internal register and is also available to the user system through I/O pins.

All 11 registers contained in the 8X360 can be accessed through standard 8X305 addressing techniques. If exceptionally high performance is required, the device is designed to take full advantage of "extended microcode" techniques frequently used with the 8X300 Family.

8X360 BLOCK DIAGRAM


$\overline{IV}$ BUS CONTROL
$\overline{IV0}$-$\overline{IV7}$
4
8
RS0-RS3
4
$\overline{IV}$ BUS INTERFACE AND REGISTER ADDRESS LOGIC
CLK
PROGRAMMABLE DIVIDER
COMMAND AND STATUS
ABORT
TRANSFER COMPLETE
ADDRESS GENERATION LOGIC, ADDRESS REGISTERS
THREE-STATE CONTROL
ADDRESS BUFFERS
16
A_0-A_{15}

VALVO

8-BIT LATCHED BIDIRECTIONAL I/O PORT 8X371

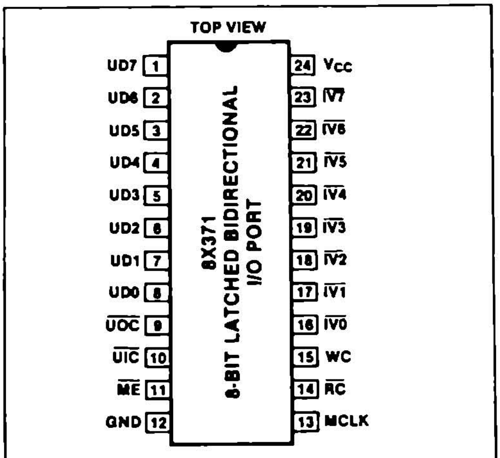
PRODUCT DESCRIPTION

The 8X371 I/O Port is a bidirectional device designed for use as an interface element in systems that use TTL-compatible busses. Typically, the 8X371 is used with the 8X305 MicroController and its associated Interface Vector ($\overline{\text{IV}}$) bus; however, it can also be used with the 8X300 MicroController or an equivalent microprocessor. The 8X371 is functionally the same and pin-for-pin compatible with the older 8T31/8X31 but features improved performance and increased drive current. As shown in the logic diagram of Figure 1, the 8X371 consists of eight identical data latches—bits 0 through 7. The latches are accessed from either of two 8-bit busses—the MicroController ($\overline{\text{IV}}$ bus) and the user data (UD) bus. Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time; in such situations, the user bus always has priority. A Master Enable ($\overline{\text{ME}}$) input is available for additional control over the $\overline{\text{IV}}$ bus. The data latches are transparent, in that, while either bus is enabled for input, all input-data transitions are propagated to the other bus, if enabled for output.

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user-bus priority for data entry)**
- **User data input synchronous with respect to MCLK**
- **Three-state TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 (or 8X300) MicroControllers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

8X371 PACKAGE and PIN DESIGNATIONS



PIN. NO.	IDENTIFIER	FUNCTION
1-8	UD7-UD0	Three-state bidirectional User Data (UD) bus. UD0 and corresponds to $\overline{\text{IV0}}$
9	$\overline{\text{UOC}}$	User Output Control—active low input to enable data output to UD0-UD7
10	$\overline{\text{UIC}}$	User Input Control—active low input to enable data input from UD0-UD7
11	$\overline{\text{ME}}$	Master Enable—active low input to enable the $\overline{\text{IV}}$ bus for data input, or data output. UD-bus operations are unaffected
12	GND	Ground
13	MCLK	Master Clock—active high input from MicroController used to strobe data into data latches from the $\overline{\text{IV}}$ and UD buses
14	$\overline{\text{RC}}$	Read Control—active low input to enable data output to $\overline{\text{IV0-IV7}}$
15	WC	Write Command—active high input from MicroController to enable the writing of data into the data latches from the $\overline{\text{IV}}$ bus provided $\overline{\text{UIC}}$ is not low
16-23	$\overline{\text{IV0-IV7}}$	Interface Vector (IV) Input/Output Bus — three-state bidirectional MicroController data bus. $\overline{\text{IV0}}$ corresponds to UD0
24	Vcc	+5V power supply

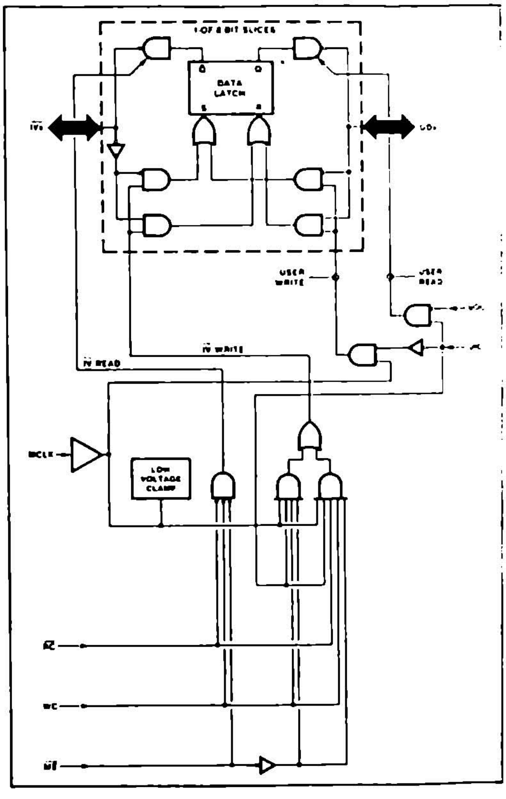


Figure 1. Logic Diagram for 8X371 I/O Port

JANUARY 1982

8-BIT LATCHED BIDIRECTIONAL I/O PORT

8X371

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus interface is controlled by the $\overline{\text{UIC}}$ and $\overline{\text{UOC}}$ inputs. Data input to the UD bus is synchronous with MCLK, that is, with $\overline{\text{UIC}}$ low, information is written into the data latches only when MCLK is high. Output drivers on the UD bus are enabled when $\overline{\text{UOC}}$ is low and $\overline{\text{UIC}}$ is high.

Table 1. Input/Output Control of UD Bus

$\overline{\text{UIC}}$	$\overline{\text{UOC}}$	MCLK	FUNCTION OF UD BUS
H	L	X	Output data
L	X	H	Input data
L	X	L	Inactive
H	H	X	Inactive

X = don't care

Table 2. Input/Output Control of $\overline{\text{IV}}$ Bus

$\overline{\text{ME}}$	$\overline{\text{RC}}$	WC	MCLK	$\overline{\text{UIC}}$	FUNCTION OF IV BUS
L	L	L	X	X	Output Data
L	X	H	H	H	Input Data
L	H	L	X	X	Inactive
L	X	H	X	L	Inactive
L	X	H	L	H	Inactive
H	X	X	X	X	Inactive

$\overline{\text{IV}}$ Bus Control

Input/output control of the $\overline{\text{IV}}$ bus is shown in Table 2, this bus is controlled by $\overline{\text{RC}}$, WC, $\overline{\text{ME}}$, and MCLK. The $\overline{\text{IV}}$ bus is enabled for output (MicroController read operation) when $\overline{\text{ME}}$, $\overline{\text{RC}}$, and WC are all low. Data is written into the data latches from the $\overline{\text{IV}}$ bus when $\overline{\text{ME}}$ is low and both WC and MCLK are high. To avoid data-input conflicts, inputs from the $\overline{\text{IV}}$ bus are inhibited when $\overline{\text{UIC}}$ is low; under all other conditions, the $\overline{\text{IV}}$ and UD busses operate independently. The MicroController Left Bank ($\overline{\text{LB}}$) and Right Bank ($\overline{\text{RB}}$) outputs can control the $\overline{\text{ME}}$ inputs for two banks of I/O devices, thus acting as a ninth address bit. If more than one I/O Port (including the addressable parts—8X372, 8X376, 8X382, etc.) are to be connected to the same bank ($\overline{\text{LB}}$ or $\overline{\text{RB}}$) of the MicroController, selection of each 8X371 must be accomplished with external control logic to avoid bus conflicts.

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note. *A logic "1" in MicroController software corresponds to a high level on the UD bus even though the $\overline{IV}$ bus is inverted.*) The 8X382 wakes up in the unselected state with all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

DC ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
T_{STG}	Storage temperature range	-65 to +150	°C

COMMERCIAL: 4.75V $\leq V_{CC} \leq$ 5.25V, 0°C $\leq T_A \leq$ 70°C
MILITARY: 4.5V $\leq V_{CC} \leq$ 5.5V, -55°C $\leq T_C \leq$ 125°C

PARAMETER		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min; I_I = -10mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max; V_{IH} = 2.7V		5	100		5	100	μA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max; V_{IL} < 0.5V		-350	-550		-350	-550	μA
V_{OL}	Low Level Output Voltage $\overline{\text{IV}}$ Bus ($\overline{\text{IV0}}$-$\overline{\text{IV7}}$)	V_{CC} = Min; I_{OL} = 16mA			0.55			0.55	V
	User Bus (UD4-UD7)	V_{CC} = Min; I_{OL} = 24mA			0.55			0.55	V
V_{OH}	High Level Output Voltage	V_{CC} = Min; I_{OH} = -3.2mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[3] $\overline{\text{IV}}$ Bus ($\overline{\text{IV0}}$-$\overline{\text{IV7}}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD4-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max; $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = V_{CC}		90	150		90	150	mA

Notes

1 The input current includes the Three-state leakage current of the output driver on the data lines
2 Only one output may be shorted at a time

AC ELECTRICAL CHARACTERISTICS

COMMERCIAL 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
MILITARY 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_C ≤ 125°C

LOADING See TEST LOADING CIRCUITS

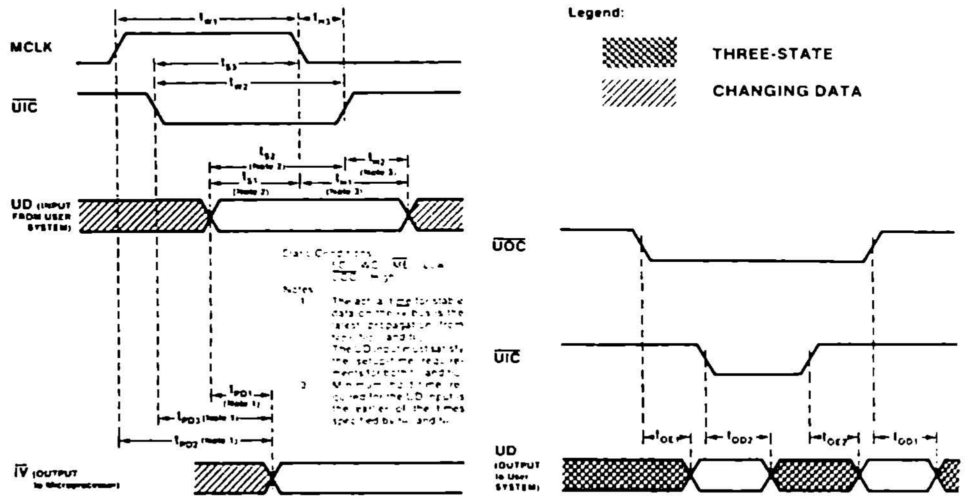
PARAMETER		REFERENCES FROM	REFERENCES TO	TEST CONDITIONS	LIMITS (COMMERCIAL) Min	Typ	Max	LIMITS (MILITARY) Min	Typ	Max	UNIT
Pulse Widths:											
t_{W1}	Clock High	↑MCLK	↓MCLK		35			35			ns
t_{W2}	User Input Control	↓$\overline{UIC}$	↑$\overline{UIC}$	MCLK = High	35			35			ns
Propagation Delays:											
t_{PD1}	UD Propagation Delay	UD	$\overline{IV}$	MCLK = High RC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			30			30	ns
t_{PD2}	UD Clock Delay	↑MCLK	$\overline{IV}$	UD Stable RC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50			50	ns
t_{PD3}	UD Input Delay	↓$\overline{UIC}$	$\overline{IV}$	UD Stable, MCLK = High, RC = WC = ME = Low			50			50	ns
t_{PD4}	$\overline{IV}$ Data Propagation Delay	$\overline{IV}$	UD	MCLK = WC = $\overline{UIC}$ = High, ME = UOC = $\overline{RC}$ = Low			45			45	ns
t_{PD5}	$\overline{IV}$ Data Clock Delay	↑MCLK	UD	WC = $\overline{UIC}$ = High, $\overline{IV}$ = Stable, ME = UOC = $\overline{RC}$ = Low			55			55	ns
Output Enable Timing:											
t_{OE1}	UD Output Enable	↓UOC	UD	$\overline{UIC}$ = High			30			30	ns
t_{OE2}	UD Input Recovery	↑$\overline{UIC}$	UD	$\overline{UOC}$ = Low			30			30	ns
t_{OE3}	$\overline{IV}$ Data Master Enable	↓$\overline{ME}$	$\overline{IV}$	WC = $\overline{RC}$ = Low			22			25	ns
t_{OE4}	$\overline{IV}$ Data Read Enable	↓$\overline{RC}$	$\overline{IV}$	WC = $\overline{ME}$ = Low			25			25	ns
t_{OE5}	$\overline{IV}$ Data Write Recovery	↓WC	$\overline{IV}$	$\overline{RC}$ = $\overline{ME}$ = Low			25			25	ns
Output Disable Timing:											
t_{OD1}	UD Output Disable	↑$\overline{UOC}$	UD	$\overline{UIC}$ = High			25			25	ns
t_{OD2}	UD Input Override	↓$\overline{UIC}$	UD	$\overline{UOC}$ = Low			30			30	ns
t_{OD3}[1]	$\overline{IV}$ Data Master Disable	↑$\overline{ME}$	$\overline{IV}$	WC = $\overline{RC}$ = Low			20			20	ns
t_{OD4}[1]	$\overline{IV}$ Data Read Disable	↑$\overline{RC}$	$\overline{IV}$	WC = $\overline{ME}$ = Low			20			20	ns
t_{OD5}[1]	$\overline{IV}$ Data Write Override	↑WC	$\overline{IV}$	$\overline{RC}$ = $\overline{ME}$ = Low			20			20	ns
Setup Times:											
t_{S1}	UD Clock Setup Time	UD	↓MCLK	$\overline{UIC}$ = Low	15			15			ns
t_{S2}	UD Setup Time	UD	↑UIC	MCLK = High	15			15			ns
t_{S3}	User Input Control Setup Time	↓$\overline{UIC}$	↓MCLK		25			25			ns
t_{S4}	$\overline{IV}$ Data Setup Time	$\overline{IV}$	↓MCLK	WC = $\overline{UIC}$ = High, $\overline{ME}$ = Low	35			35			ns
t_{S5}	$\overline{IV}$ Master Enable Setup Time	↓$\overline{ME}$	↓MCLK	WC = $\overline{UIC}$ = High	30			30			ns
t_{S6}	$\overline{IV}$ Write Control Setup Time	↑WC	↓MCLK	$\overline{ME}$ = Low, $\overline{UIC}$ = High	30			30			ns

AC ELECTRICAL CHARACTERISTICS (Cont'd)

PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
	FROM	TO		Min	Typ	Max	Min	Typ	Max	
Hold Times:										
t_{H1} UD Clock Hold Time	↓MCLK	UD	$\overline{UIC}$ = Low	15			15			ns
t_{H2} UD Control Hold Time	↑$\overline{UIC}$	UD	MCLK = High	15			15			ns
t_{H3} User Input Control Hold Time	↓MCLK	↑$\overline{UIC}$		0			0			ns
t_{H4} $\overline{IV}$ Data Hold Time	↓MCLK	$\overline{IV}$	WC = $\overline{UIC}$ = High, $\overline{ME}$ = Low	5			5			ns
t_{H5}[2] $\overline{IV}$ Master Enable Hold Time	↓MCLK	↑$\overline{ME}$	WC = $\overline{UIC}$ = High	0			0			ns
t_{H6} $\overline{IV}$ Write Control Hold Time	↓MCLK	↓WC	$\overline{ME}$ = Low, $\overline{UIC}$ = High	0			0			ns

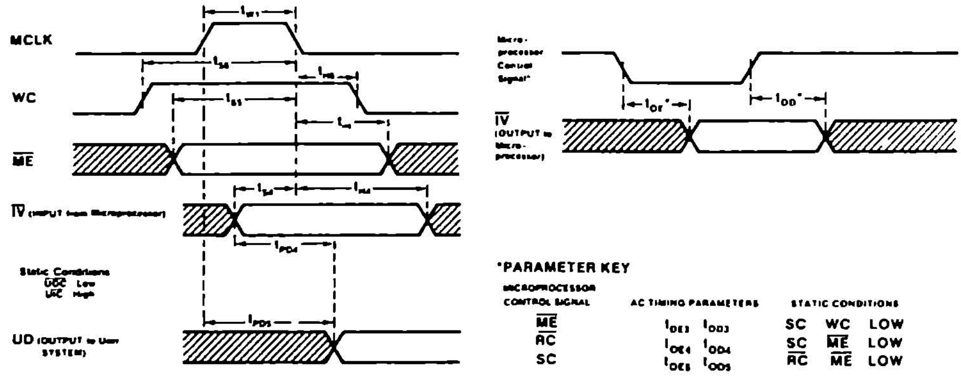
Notes

1. These parameters are measured with a capacitive loading of 50 pF and represent the output driver turn-off time
2. If $\overline{ME}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port


MCLK
UIC
UD (INPUT FROM USER SYSTEM)
IV (OUTPUT to Microprocessor)
Legend:
THREE-STATE
CHANGING DATA
UOC
UIC
UD (OUTPUT to User SYSTEM)

a. User Data Input Timing

b. User Data Output Enable Timing


MCLK
WC
ME
IV (INPUT from Microprocessor)
Static Conditions
UOC Low
UIC High
UD (OUTPUT to User SYSTEM)
Micro-processor Control Signal*
IV (OUTPUT to Micro-processor)

*PARAMETER KEY

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS		STATIC CONDITIONS		
$\overline{ME}$	t_{OE3}	t_{OD3}	SC	WC	LOW
$\overline{RC}$	t_{OE4}	t_{OD4}	SC	$\overline{ME}$	LOW
SC	t_{OE5}	t_{OD5}	$\overline{RC}$	$\overline{ME}$	LOW

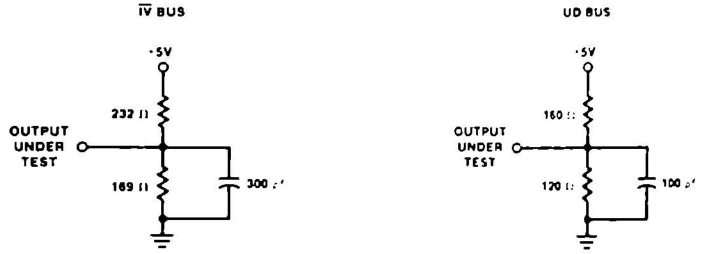
c. MicroController Write Cycle Timing

d. MicroController Select Cycle Timing

Figure 2. Timing Diagram

8-BIT LATCHED BIDIRECTIONAL I/O PORT 8X371

TEST LOADING CIRCUITS


$\overline{\text{IV}}$ BUS
+5V
232 Ω
OUTPUT UNDER TEST
169 Ω
300 pF
UD BUS
+5V
160 Ω
OUTPUT UNDER TEST
120 Ω
100 pF

ORDERING INFORMATION

COMMERCIAL—
- N8X371N (Plastic)
- N8X371I (Ceramic)

MILITARY—
- S8X371I/883B
- S8X371I/883C

8-BIT LATCHED BIDIRECTIONAL I/O PORT 8X371

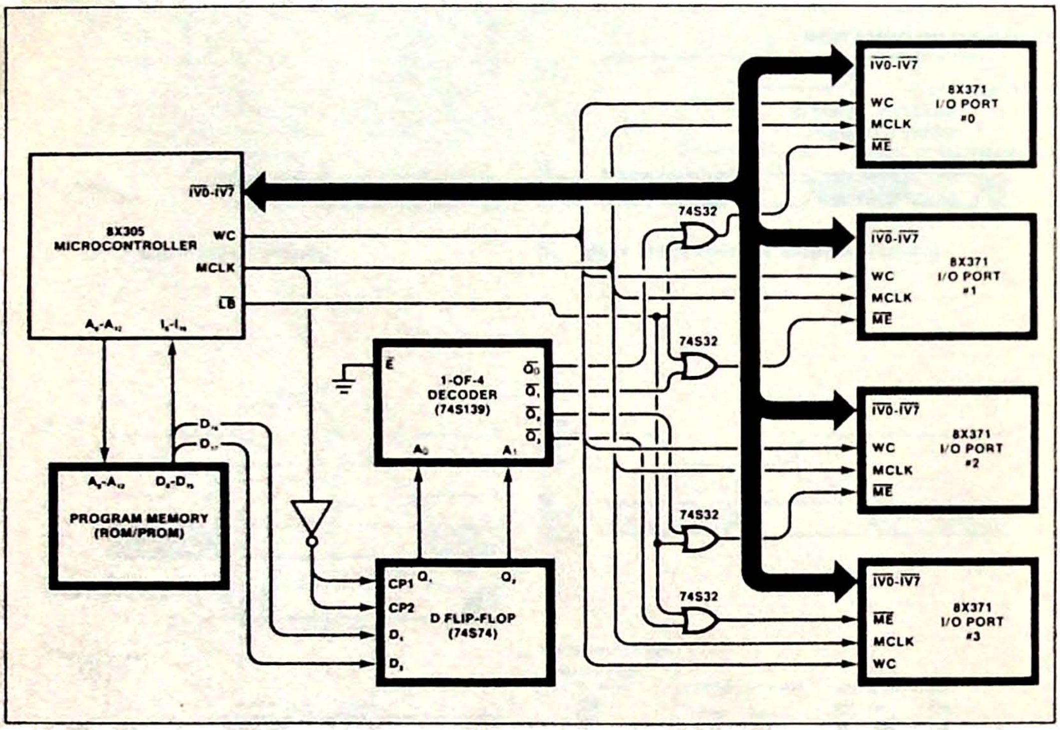
APPLICATIONS

In some applications, performance of a MicroController system can be enhanced by using the 8X371 I/O Port instead of an addressable 8X372 port. Using a technique referred to as Extended Microcode or Fast $\overline{\text{IV}}$ Select, the address select cycles which normally precede a read or write operation when using an 8X372 can be eliminated by use of the 8X371.

This technique is often used in bit slice microprocessor designs and involves widening the program memory beyond the normal 16-bit requirement of the MicroController. The extra bits are used as enable signals for the 8X371 ports. Thus, the 8X371 is enabled during the instruction cycle in which it is required for input/output operations. Since the software overhead of separate address select cycles is eliminated, the overall system performance is improved.

As shown in the accompanying diagram, the program memory is extended by two bit positions (D_{16} and D_{17}), permitting any one of four 8X371 ports to be enabled during those instructions that perform input/output operations. Because of timing considerations, latches must be used to hold the Extended Microcode through the end of the instruction cycle. A decoder is used to obtain four enable signals from the two extra bits. The decoder outputs are ORed with the $\overline{\text{LB}}$ output of the 8X305, thus, all four I/O ports are placed on the Left Bank of the $\overline{\text{IV}}$ bus.

I/O PORT SELECTION USING EXTENDED MICROCODE


8X305
MICROCONTROLLER
IV0-IV7
WC
MCLK
LB
A0-A12
I0-I15
D16
D17
A0-A12
D0-D15
PROGRAM MEMORY
(ROM/PROM)
E
1-OF-4
DECODER
(74S139)
O0
O1
O2
O3
A0
A1
CP1
CP2
D1
D2
Q1
Q2
D FLIP-FLOP
(74S74)
74S32
74S32
74S32
74S32
IV0-IV7
WC
MCLK
ME
8X371
I/O PORT
#0
IV0-IV7
WC
MCLK
ME
8X371
I/O PORT
#1
IV0-IV7
WC
MCLK
ME
8X371
I/O PORT
#2
IV0-IV7
ME
MCLK
WC
8X371
I/O PORT
#3

JANUARY 1982

ADDRESSABLE/BIDIRECTIONAL I/O PORTS 8X372/8X376

PRODUCT IDENTITY

8X372—Synchronous, three-state, bidirectional I/O port with programmed address

8X376—Asynchronous, three-state, bidirectional I/O port with programmed address

PRODUCT DESCRIPTION

Each of these I/O ports is an addressable device designed for use as a bidirectional interface element in systems that use TTL-compatible busses. Typically, these I/O ports are used with the 8X305 MicroController and its associated Interface Vector ($\overline{\text{IV}}$) bus; however, either port can also be used with the 8X300 MicroController or an equivalent microprocessor. The 8X372 and 8X376 are funtionally the same and pin-for-pin compatible with their respective counterparts, the 8T32/8X32 and 8T36/8X36; however, the new parts feature better performance, increased drive current, and improved programming procedures.

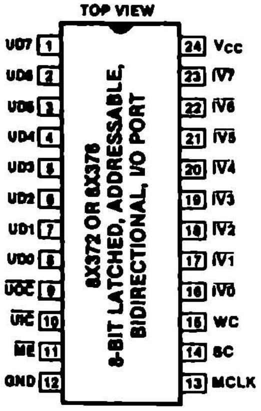
As shown in the logic diagram of Figure 1, each I/O port consists of eight identical data latches—bits 0 through 7. These latches are accessed through either of two 8-bit busses—one connecting to the MicroController ($\overline{\text{IV}}$ bus) and the other to the user system (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time. In such situations, the user bus always has priority. The data latches are transparent, in that, while either bus is enabled for input, all transitions in input data are propagated to the other bus, if enabled for output.

Both the 8X372 and 8X376 are available with preprogrammed addresses (0_{10} through 255_{10}); either device can be field-programmed over the same address range. Input/output operations can begin once the I/O port is selected and appropriate control signals are generated. Port selection is implemented by putting the I/O port address (0_{10}-255_{10}) on the $\overline{\text{IV}}$ bus; once selected, the I/O port remains selected until a different "port address" is put on the bus. Thus, software overhead is minimized. Data is accessible on the UD bus at all times. A Master Enable ($\overline{\text{ME}}$) input, which is typically connected to the Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) output of the MicroController, provides the capability of organizing the $\overline{\text{IV}}$ bus into two separate and independent banks of I/O devices.

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user-bus priority for data entry)**
- **User data input synchronous (8X372) or asynchronous (8X376) with respect to MCLK**
- **Programmed MicroController port address**
- **Three-state TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 or 8X300 MicroControllers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

8X372/8X376 PACKAGE and PIN DESIGNATIONS



PIN. NO.	IDENTIFIER	FUNCTION
1-8	UD7-UD0	Three-state bidirectional User Data (UD) bus. UD0 corresponds to $\overline{\text{IV0}}$
9	$\overline{\text{UOC}}$	User Output Control—active low input to enable data output to UD0-UD7
10	$\overline{\text{UIC}}$	User Input Control—active low input to enable data input from UD0-UD7
11	$\overline{\text{ME}}$	Master Enable—active low input to enable the $\overline{\text{IV}}$ bus for data input, data output or $\overline{\text{IV}}$ address selection/deselection. UD-bus operations are unaffected
12	GND	Ground
13	MCLK	Master Clock—active high input from MicroController used to strobe data into data latches from the IV bus and for the synchronous 8X372, from the UD bus. MCLK also synchronizes IV address selection
14	SC	Select Command—active high input from MicroController to enable $\overline{\text{IV}}$ address input from the $\overline{\text{IV}}$ bus for device selection
15	WC	Write Command—active high input from MicroController to enable the writing of data into the data latches from the $\overline{\text{IV}}$ bus, provided $\overline{\text{UIC}}$ is not low
16-23	$\overline{\text{IV0}}$-$\overline{\text{IV7}}$	Interface Vector (Input/Output Bus)—three-state, bidirectional, MicroController data bus. $\overline{\text{IV0}}$ corresponds to UD0
24	Vcc	+5V power supply

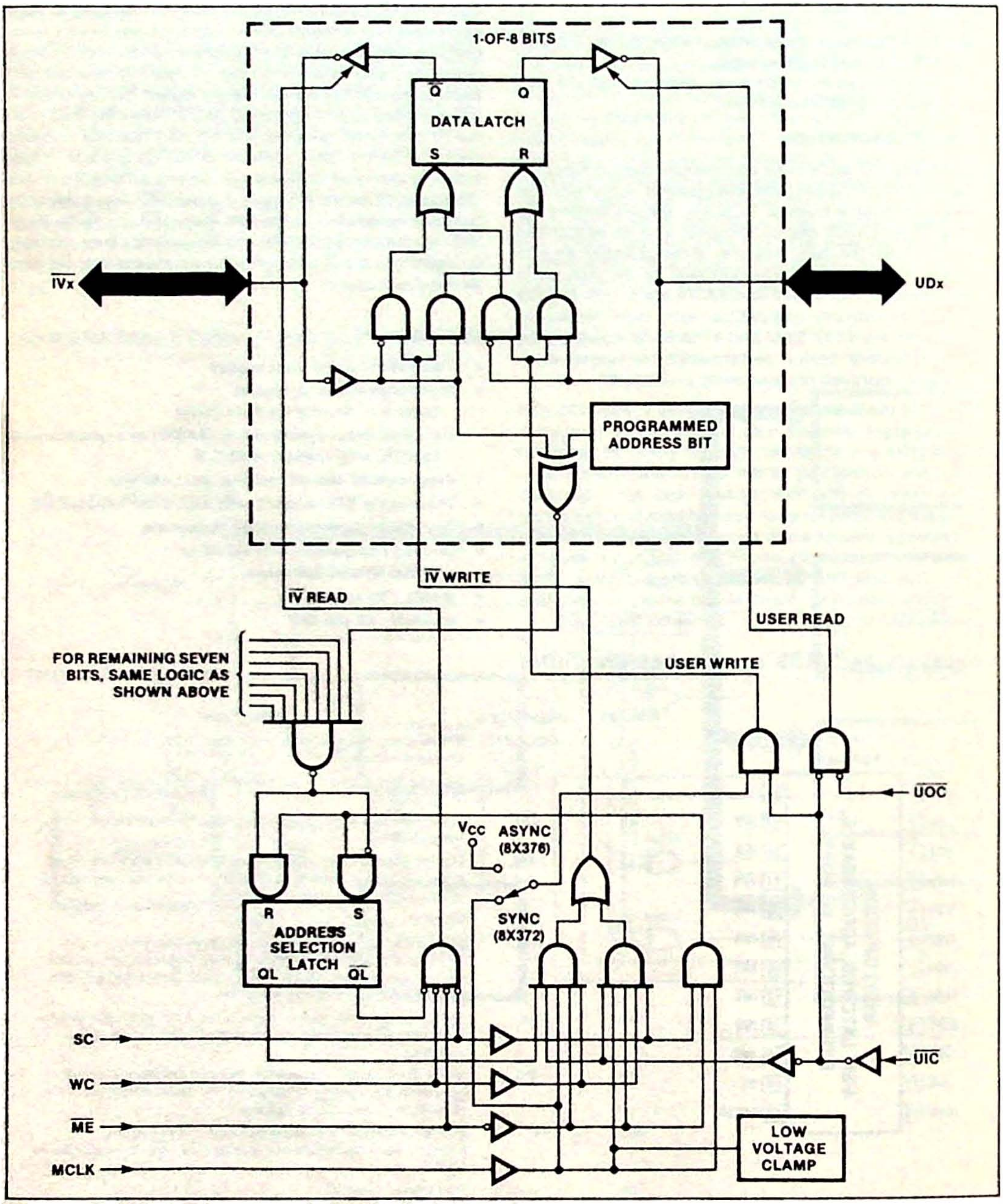

1-OF-8 BITS
$\overline{Q}$
Q
DATA LATCH
S
R
$\overline{IV}$x
UDx
PROGRAMMED ADDRESS BIT
$\overline{IV}$ WRITE
$\overline{IV}$ READ
USER READ
USER WRITE
FOR REMAINING SEVEN BITS, SAME LOGIC AS SHOWN ABOVE
$\overline{UOC}$
V$_{CC}$
ASYNC (8X376)
SYNC (8X372)
R
S
ADDRESS SELECTION LATCH
QL
$\overline{QL}$
SC
WC
$\overline{ME}$
MCLK
$\overline{UIC}$
LOW VOLTAGE CLAMP

Figure 1. Logic Diagram for 8X372/8X376 I/O Ports

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus interface is controlled by the $\overline{UIC}$ and $\overline{UOC}$ inputs. For the 8X372, data input from the UD bus is written synchronously with MCLK, that is, with $\overline{UIC}$ low, information is written into the data latches only when MCLK is high. In the case of the 8X376, data input is asynchronous, in that, with $\overline{UIC}$ low, data is latched in without regard to the level of MCLK (Note *To avoid the possibility of processor error when using the asynchronous 8X376, the $\overline{IV}$ bus should not be read during the time the data latches are changing due to user input.*) Output drivers on the UD bus are enabled when UOC is low and $\overline{UIC}$ is high.

Table 1. Input/Output Control of UD Bus

$\overline{UIC}$	$\overline{UOC}$	MCLK	FUNCTION OF UD BUS	
			8X372	8X376
H	L	X	Output data	Output data
L	X	H	Input data	Input data
L	X	L	Inactive	Input data
H	H	X	Inactive	Inactive

X = don't care

$\overline{IV}$ Bus Control

Input/output control of the $\overline{IV}$ bus is shown in Table 2; this bus is controlled by SC, WC, $\overline{ME}$, MCLK and the current state of the internal address selection latch. AS shown in Table 2, $\overline{UIC}$ is required to indicate priority of the UD bus for data input operations. The selection latch in the I/O port stores the result of the most recent $\overline{IV}$ address selection. The latch is set when the internally preprogrammed address of the port matches the address on the $\overline{IV}$ bus during an address-selection operation (SC = MCLK = High/WC = Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The $\overline{IV}$ bus can transfer data only when the selection latch is set. As shown in the APPLICATION DIAGRAM, the MicroController Left Bank ($\overline{LB}$) and Right Bank ($\overline{RB}$) outputs can control the $\overline{ME}$ inputs for two banks of I/O devices, thus, acting as a ninth address bit.

Table 2. Input/Output Control of $\overline{IV}$ Bus

$\overline{ME}$	SC	WC	MCLK	$\overline{UIC}$	SELECTION LATCH	FUNCTION OF $\overline{IV}$ BUS
L	L	L	X	X	Set	Output Data
L	L	H	H	H	Set	Input Data
L	H	L	H	X	X	Input Address*
L	H	H	H	H	X	Input data and address*
L	H	H	H	L	X	Input Address*
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

X = don't care

* Selection latch is updated

Data is written into the data latches of a selected device from the $\overline{IV}$ bus when WC, MCLK, and $\overline{UIC}$ are all high and $\overline{ME}$ is low. To prevent data-input conflicts, inputs from the $\overline{IV}$ bus are inhibited when $\overline{UIC}$ is low, under all other conditions, the $\overline{IV}$ and UD busses operate independently. Output drivers on the $\overline{IV}$ bus of a selected device are enabled when $\overline{ME}$, WC, and SC are all low and the address selection latch is set. With SC and WC both high (shaded entry of Table 2), the bit pattern present on $\overline{IV0}$-$\overline{IV7}$ is interpreted as both input data and $\overline{IV}$ address. Provided $\overline{UIC}$ is high, the data is latched into the data latches whether or not the I/O port has been previously selected. If the preprogrammed address of the I/O port matches the bit pattern on $\overline{IV0}$-$\overline{IV7}$ when SC and WC are both high, the selection latch is set, otherwise, it is reset. Note *The MicroController never drives both SC and WC high at the same time.*

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note *A logic "1" in MicroController software corresponds to a high level on the UD bus even though the $\overline{IV}$ bus is inverted.*) Both the 8X372 and 8X376 wakeup with the address selection latch in the unselected state and all data bits latched at the logic 1" level. UD bus outputs high if enabled.

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

Both 8X372 and 8X376 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level ($\leq$0.8V) inputs on all $\overline{IV}$ bus lines (address 255_{10}) will select the device. To program a given address bit to match a high level ($\geq$2.0V) input on the corresponding $\overline{IV}$ pin (a logical "0" to the MicroController), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state—$\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = +5V and SC = WC = MCLK = GND; leave the UD and $\overline{IV}$ bus pins open

Step 2 Increase Vcc to 9.0 volts.

Step 3 After Vcc has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level $\overline{IV}$ address bit. The I/O port is programmed from the user bus (UD0-UD7) for addressing from the MicroController bus ($\overline{IV0}$-$\overline{IV7}$).

Step 4: Return V_{CC} to 0-volts. Note *If the programming of all address bits is completed in less than 1-second, V_{CC} can remain at 9.0-volts for the required interval of time.*

Step 5 Step 1 through 3 are applicable to the programming of each address bit that requires a high-level $\overline{IV}$ match

Table 3. Programming Specifications

PARAMETERS	LIMITS Min	LIMITS Typ	LIMITS Max	UNITS
V_{CCP} -- Programming supply voltage Address	8.75	9.0	9.25	V
Protect		0		V
Maximum time V_{CCP} > 5.25V			1.0	Sec
Programming voltage Address	8.75	9.0	9.25	V
Protect	13.5		14.0	V
Programming current Address			5	mA
Protect			75	mA
t_r -- Programming pulse rise time Address	0.1		1.0	μS
Protect	0.1			μS
t_w - Programming pulse width	0.5		1.0	μS

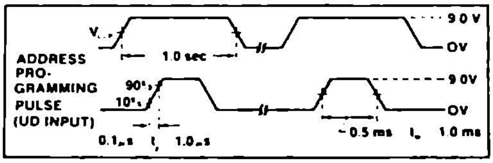


Figure 2. Address Programming Pulse

Step 6 To verify that the address is properly programmed, return V_{CC} to +5V, set $\overline{IV0}$-$\overline{IV7}$ to the desired inverted binary address pattern, set $\overline{ME}$ = WC = Low and SC = MCLK = High. If there are no programming errors, subsequent data written from $\overline{IV0}$-$\overline{IV7}$ (WC = High) will appear inverted on UD0-UD7.

Address Protect

After programming the I/O Port, steps should be taken to isolate the address circuits and make these circuits permanently immune to further change.

Step 1 Set V_{CC} and all control inputs to 0-volts (V_{CC} = $\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = SC = WC = MCLK = 0V), $\overline{IV0}$-$\overline{IV7}$ open circuit.

Step 2 Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0-UD7)—refer to Table 3 for min/max specifications pertaining to voltage and current.

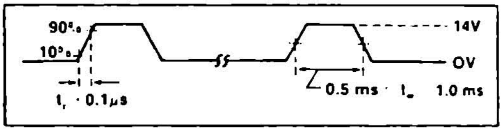


Figure 3. Protect Programming Pulse

Step 3 Verify that the address circuits for each bit is isolated by applying 7-volts, in turn, to each user-bus pin (UD0-UD7) and measuring less than 1-milliampere of input current. Note: *Setup conditions are the same as those in Step 1, the rise time of verification voltage must be 100-microseconds or slower.*

DC ELECTRICAL CHARACTERISTICS

COMMERCIAL 4.75V ≤ V_{CC} ≤ 5.25V 0°C ≤ T_A ≤ 70°C
MILITARY 4.5V ≤ V_{CC} ≤ 5.5V -55°C ≤ T_C ≤ 125°C

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage[1]	+7	Vdc
V_{IN}	Input voltage[3]	+5.5	Vdc
T_{STG}	Storage temperature range	-65 to +150	°C

PARAMETER		TEST CONDITIONS	LIMITS (COMMERCIAL) Min	LIMITS (COMMERCIAL) Typ	LIMITS (COMMERCIAL) Max	LIMITS (MILITARY) Min	LIMITS (MILITARY) Typ	LIMITS (MILITARY) Max	UNIT
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min, I_I = -10mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max, V_{IH} = 2.7V		5.0	100		5.0	100	μA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max, V_{IL} = 0.5V		-350	-550		-350	-550	μA
V_{OL}	Low Level Output Voltage $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Min, I_{OL} = 16mA			0.55		0.55	V	
	User Bus (UD0-UD7)	V_{CC} = Min, I_{OL} = 24mA			0.55			0.55	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = -3.2mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[2] $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD0-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max, $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150		90	150	mA

Notes

1. The input current includes the Three-state leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.
3. These limits do not apply during address programming.

AC ELECTRICAL CHARACTERISTICS

COMMERCIAL 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
MILITARY 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_C ≤ 125°C
LOADING See TEST LOADING CIRCUITS

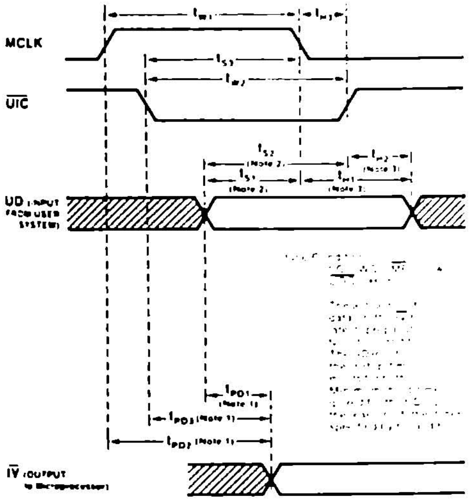
PARAMETER	REFERENCES[1]		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
	FROM	TO		Min	Typ	Max	Min	Typ	Max	
Pulse Widths: t_{PW1} Clock High	↑MCLK	↓MCLK		35			35			ns
t_{PW2} User Input Control	↓$\overline{\text{UIC}}$	↑$\overline{\text{UIC}}$	MCLK = High	35			35			ns
Propagation Delays: t_{PD1} UD Propagation Delay	UD	$\overline{\text{IV}}$	MCLK = High SC = WC = $\overline{\text{ME}}$ = $\overline{\text{UIC}}$ = Low			30			30	ns
t_{PD2} UD Clock Delay (8X792 only)	↑MCLK	$\overline{\text{IV}}$	UD = Stable SC = WC = $\overline{\text{ME}}$ = $\overline{\text{UIC}}$ = Low			50			50	ns
t_{PD3} UD Input Delay	↓$\overline{\text{UIC}}$	$\overline{\text{IV}}$	UD = Stable, MCLK = High SC = WC = $\overline{\text{ME}}$ = Low			50			50	ns
t_{PD4} $\overline{\text{IV}}$ Data Propagation Delay	$\overline{\text{IV}}$	UD	MCLK = WC = $\overline{\text{UIC}}$ = High, $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = SC = Low			45			45	ns
t_{PD5} $\overline{\text{IV}}$ Data Clock Delay	↑MCLK	UD	WC = $\overline{\text{UIC}}$ = High, $\overline{\text{IV}}$ = Stable, $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = SC = Low			55			55	ns
Output Enable Timing: t_{OE1} UD Output Enable	↓$\overline{\text{UOC}}$	UD	$\overline{\text{UIC}}$ = High			30			30	ns
t_{OE2} UD Input Recovery	↑$\overline{\text{UIC}}$	UD	$\overline{\text{UOC}}$ = Low			30			30	ns
t_{OE3} $\overline{\text{IV}}$ Data Master Enable	↓$\overline{\text{ME}}$	$\overline{\text{IV}}$	WC = SC = Low			22			25	ns
t_{OE5} $\overline{\text{IV}}$ Data Write Recovery	↓WC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			25			25	ns
t_{OE6} $\overline{\text{IV}}$ Data Select Recovery	↓SC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			25			25	ns
Output Disable Timing: t_{OD1} UD Output Disable	↑$\overline{\text{UOC}}$	UD	$\overline{\text{UIC}}$ = High			25			25	ns
t_{OD2} UD Input Override	↓$\overline{\text{UIC}}$	UD	$\overline{\text{UOC}}$ = Low			30			30	ns
t_{OD3}[2] $\overline{\text{IV}}$ Data Master Disable	↑$\overline{\text{ME}}$	$\overline{\text{IV}}$	WC = SC = Low			20			20	ns
t_{OD4}[2] $\overline{\text{IV}}$ Data Write Override	↑WC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			20			20	ns
t_{OD5}[2] $\overline{\text{IV}}$ Data Select Override	↑SC	$\overline{\text{IV}}$	WC = $\overline{\text{ME}}$ = Low			20			20	ns
Setup Times: t_{S1} UD Clock Setup Time (8X372 only)	UD	↓MCLK	$\overline{\text{UIC}}$ = Low	15			15			ns
t_{S2} UD Control Setup Time	UD	↑$\overline{\text{UIC}}$	MCLK = High	15			15			ns
t_{S3} User Input Control Setup Time (8X372 only)	↓$\overline{\text{UIC}}$	↓MCLK		25			25			ns
t_{S4} $\overline{\text{IV}}$ Data Setup Time	$\overline{\text{IV}}$	↓MCLK	WC = High or SC = High, $\overline{\text{ME}}$ = Low; $\overline{\text{UIC}}$ = High	35			35			ns
t_{S5}[3] $\overline{\text{IV}}$ Master Enable Setup Time	↓ME	↓MCLK	WC = High or SC = High; UIC = High	30			30			ns
t_{S6} $\overline{\text{IV}}$ Write Control Setup Time	↑WC	↓MCLK	SC = $\overline{\text{ME}}$ = Low, $\overline{\text{UIC}}$ = High	30			30			ns

AC ELECTRICAL CHARACTERISTICS (Cont'd)

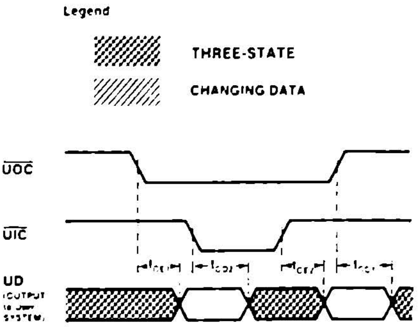
PARAMETER		REFERENCES		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
		FROM	TO		Min	Typ	Max	Min	Typ	Max	
t_{S7}	$\overline{IV}$ Select Control Setup Time	↑SC	↓MCLK	WC $\overline{ME}$ Low	30			30			ns
Hold Times:											
t_{H1}	UD Clock Hold Time (8X372 only)	↓MCLK	UD	$\overline{UIC}$ Low	15			15			ns
t_{H2}	UD Control Hold Time	↑$\overline{UIC}$	UD	MCLK High	15			15			ns
t_{H3}	User Input Control Hold Time (8X372 only)	↓MCLK	↑$\overline{UIC}$		0			0			ns
t_{H4}	$\overline{IV}$ Data Hold Time	↓MCLK	$\overline{IV}$	$\overline{WC}$ High or SC High $\overline{ME}$ Low $\overline{UIC}$ High	5			5			ns
t_{H5}[3]	$\overline{IV}$ Master Enable Hold Time	↓MCLK	↑$\overline{ME}$	$\overline{WC}$ High or SC High, $\overline{UIC}$ High	0			0			ns
t_{H6}	$\overline{IV}$ Write Control Hold Time	↓MCLK	↓WC	SC $\overline{ME}$ Low $\overline{UIC}$ High	0			0			ns
t_{H7}	$\overline{IV}$ Select Control Hold Time	↓MCLK	↓SC	WC $\overline{ME}$ Low	0			0			ns

Notes

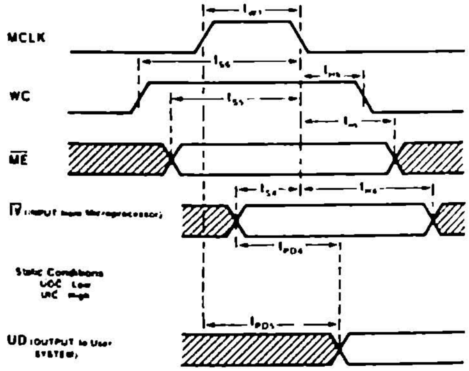
1. All measurements to the $\overline{IV}$ bus assumes the address selection latch is set
2. These parameters are measured with a capacitive loading of 50pf and represent the output driver turn-off time
3. If $\overline{ME}$ is to be high (inactive) it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing [illegible] of the I/O port


MCLK
$\overline{UIC}$
UD (INPUT FROM USER SYSTEM)
$\overline{IV}$ (OUTPUT to Microprocessor)
t_{W1}
t_{H1}
t_{S1}
t_{W2}
t_{S2} (Note 2)
t_{H2} (Note 1)
t_{S1} (Note 2)
t_{H1} (Note 1)
t_{PD1} (Note 1)
t_{PD3} (Note 1)
t_{PD2} (Note 1)

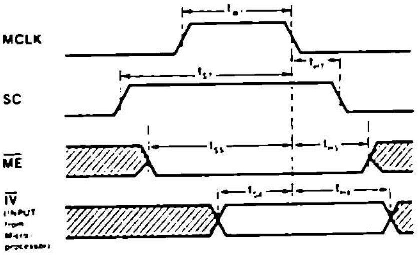
a. User Data Input Timing


Legend
THREE-STATE
CHANGING DATA
$\overline{UOC}$
$\overline{UIC}$
UD (OUTPUT to User SYSTEM)
t_{CE1}
t_{CD2}
t_{CE2}
t_{CD1}

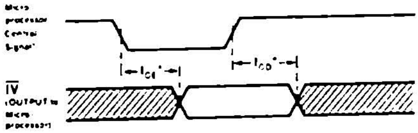
b. User Data Output Enable


MCLK
WC
$\overline{ME}$
$\overline{IV}$ (INPUT from Microprocessor)
Static Conditions
UOC Low
UIC High
UD (OUTPUT to User SYSTEM)
t_{W3}
t_{S6}
t_{H5}
t_{S5}
t_{H5}
t_{S4}
t_{H4}
t_{PD4}
t_{PD5}

c. MicroController Write Cycle Timing


MCLK
SC
$\overline{ME}$
$\overline{IV}$ (INPUT from Micro-processor)
t_{W}
t_{S7}
t_{H7}
t_{S5}
t_{H5}
t_{S4}
t_{H4}

d. MicroController Select Cycle Timing


Micro-processor Control Signal*
$\overline{IV}$ (OUTPUT to Micro-processor)
t_{CE}*
t_{CD}*

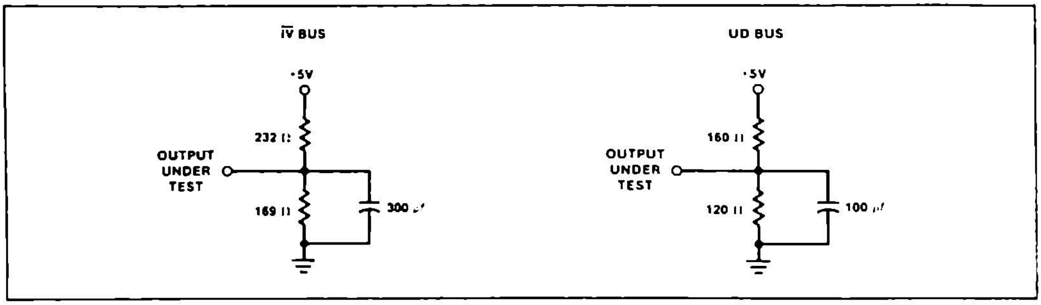
*PARAMETER KEY

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
$\overline{ME}$	t_{OE1} t_{OD1}	SC WC LOW
WC	t_{OE5} t_{OD5}	SC $\overline{ME}$ LOW
SC	t_{OE6} t_{OD6}	WC $\overline{ME}$ LOW

e. MicroController Output Enable Timing

Figure 2. Timing Diagrams

TEST LOADING CIRCUITS



ORDERING INFORMATION

Preprogrammed Addresses

Both the 8X372 and the 8X376 can be ordered with addresses preprogrammed at the factory. To order these parts, use the following part-number format.

COMMERCIAL—

N8X372N-zzz or N8X376N-zzz Plastic
N8X372I-zzz or N8X376I-zzz Ceramic

MILITARY—

S8X372I/883B-zzz or S8X376I/883B-zzz
S8X372I/883C-zzz or S8X376I/883C-zzz

where, zzz=any address from 000_{10} through 255_{10}. (Note. *I/O Ports with preprogrammed addresses 0_{10} through 15_{10} are stock items; consult the nearest Signetics Sales and Service Office when ordering parts in the address range of 16_{10} through 25510.*)

Field-Programmable Addresses

Relevant ordering information for these parts is given below; unless otherwise indicated, referenced packaging data and operating specifications for parts with preassigned or field-programmable addresses are the same

COMMERCIAL—

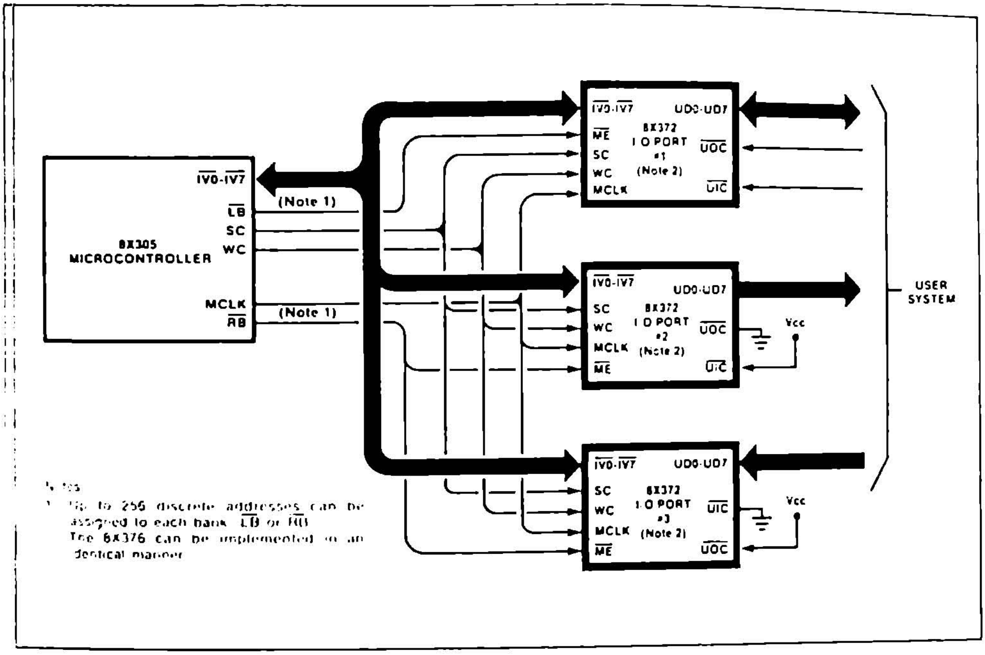
N8X372N or N8X376N Plastic
N8X372I or N8X376I Ceramic

MILITARY—

S8X372I/883B or S8X372I/883C
S8X376I/883C or S8X376I/883C

APPLICATIONS

One way of using I/O Ports in a microprocessor-based system is shown in the following application diagram. there are many other ways of implementing I/O functions with these parts, both singly and in combination. By proper control of the $\overline{UIC}$ and $\overline{UOC}$ lines, the user can implement bidirectional data transfers, exercise system control and/or read system status. In the concept shown here, I/O Port #1 is setup for bidirectional data transfers and I/O Ports #2 and #3, respectively, serve as dedicated output and input devices.


IV0-IV7
LB
SC
WC
MCLK
RB
8X305
MICROCONTROLLER
(Note 1)
(Note 1)
IV0-IV7
UD0-UD7
ME
SC
WC
MCLK
8X372
I/O PORT
#1
(Note 2)
UOC
UIC
IV0-IV7
UD0-UD7
SC
WC
MCLK
ME
8X372
I/O PORT
#2
(Note 2)
UOC
UIC
Vcc
IV0-IV7
UD0-UD7
SC
WC
MCLK
ME
8X372
I/O PORT
#3
(Note 2)
UIC
UOC
Vcc
USER
SYSTEM

Notes:

1. Up to 256 discrete addresses can be assigned to each bank $\overline{LB}$ or $\overline{RB}$.
2. The 8X376 can be implemented in an identical manner.

ADDRESSABLE/BIDIRECTIONAL I/O PORT WITH PARITY 8X374

FEATURES

- TWO BIDIRECTIONAL BUSSES
- INDEPENDENT BUS OPERATION
- PARITY CHECKING AND GENERATION FOR USER SYSTEM BUS
- USER SELECTABLE EVEN OR ODD PARITY SUPPORT
- PROGRAMMABLE WITH 8X305 MICROCONTROLLER PORT ADDRESS
- POWER UP TO PREDETERMINED LOGIC STATE
- THREE-STATE TTL OUTPUTS AND HIGH DRIVE CAPABILITIES
- SINGLE +5 VOLT POWER SUPPLY
- 28-PIN DUAL IN-LINE PACKAGE

PRODUCT DESCRIPTION

The 8X374 I/O Port with parity provides all of the circuitry required to easily interface the 8X300 MicroController Family to any user system that has high data integrity requirements and demands the use of parity in the system design. It is an 8-bit, bidirectional device that operates synchronously with the MCLK of the MicroController.

Operationally, the device is identical to the 8X372 I/O Port, except that additional circuitry is included to enable parity checking for input data to the 8X305 and parity generation on output data. Each pin functions the same as its counterpart on the 8X372, with four additional pins added to support the parity capabilities. These pins have the following functions:

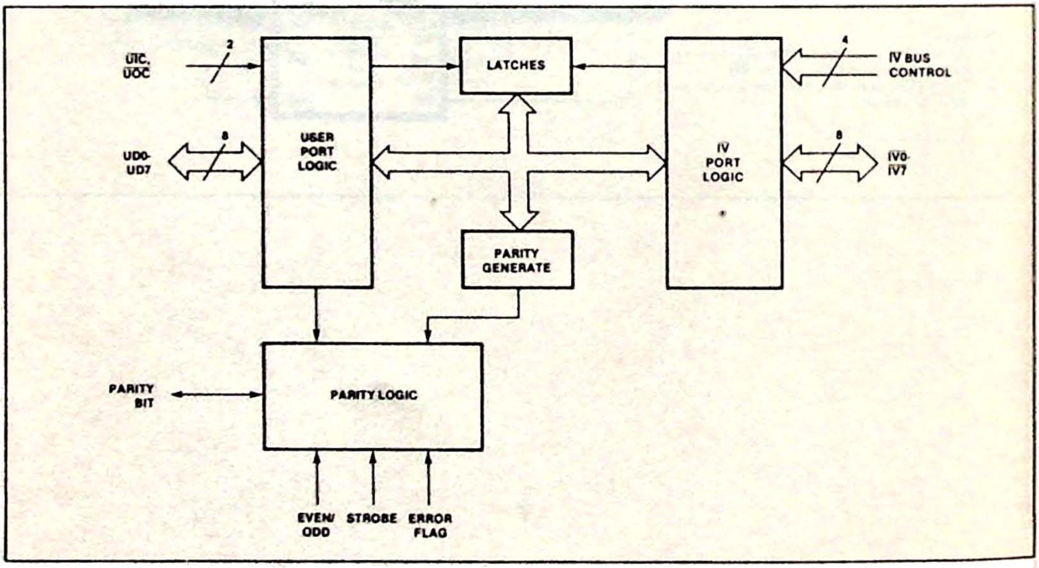
- Parity Bit (input and output)
- Parity Select (odd or even)
- Parity Error Flag
- Error Flag Strobe

When the 8X374 is being used to input data to the MicroController, it reads 9 bits from the user bus, checks parity, and transmits 8 bits to the Interface Vector ($\overline{IV}$) Bus of the 8X305. If a parity error is detected, an error flag is latched into the device that can be tested either immediately or at any later point prior to another input cycle. The flag remains set until the error flag strobe is pulsed.

Conversely, on an output cycle, the 8X374 accepts 8 bits of data from the MicroController, generates parity, and outputs 9 bits onto the user bus.

Use of odd or even parity is at user option and selectable through a pin on the device.

8X374 PARITY I/O PORT



4-INPUT/4-OUTPUT ADDRESSABLE I/O PORT 8X382

PRODUCT DESCRIPTION

The 8X382 I/O Port is an addressable, three-state device designed for use as an interface element in systems that use TTL-compatible busses. Typically, the 8X382 is used with the 8X305 MicroController and its associated Interface Vector $\overline{IV}$ bus; however, it can also be used with the 8X300 MicroController or an equivalent microprocessor. The 8X382 is functionally the same and pin-for-pin compatible with the older 8X42; however, the new port features better performance, increased drive current, and improved programming procedures.

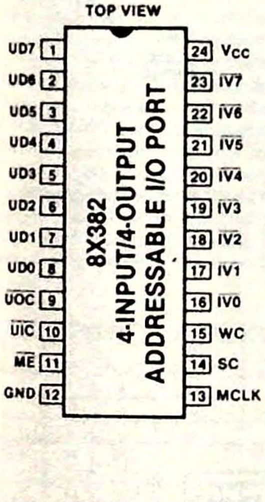
As shown in the logic diagram of Figure 1, the I/O port consists of eight data latches—bits 0 through 7. These latches are accessed through either of two busses—an 8-bit bidirectional $\overline{IV}$ bus connected to the MicroController and a User Data UD bus consisting of four dedicated inputs bits UD0 through UD3 and four dedicated outputs bits UD4 through UD7. All eight bits may be read from or four data bits $\overline{IV4}$-$\overline{IV7}$ can be written into via the $\overline{IV}$ bus; eight bits of I/O address can be written from the $\overline{IV}$ bus. Separate controls are provided for each bus and both busses operate independently. The I/O data latches are transparent, in that, when either bus is enabled for input, all transitions in input data are propagated to the other bus, if that bus in enabled for output.

The 8X382 is available with preprogrammed addresses 0_{10} through 255_{10}; it can also be field-programmed over the same address range. Input/output operations can begin once the I/O port is selected and appropriate control signals are generated. Port selection is implemented by putting the I/O port address 0_{10}-255_{10} on the $\overline{IV}$ bus; once selected, the I/O port remains selected until a different port address is put on the bus. Thus, software overhead is minimized. Data is accessible on the UD bus at all times. A Master Enable $\overline{ME}$ input, which is typically connected to the Left Bank $\overline{LB}$ or Right Bank $\overline{RB}$ output of the MicroController, provides the capability of organizing the $\overline{IV}$ bus into two separate and independent banks of I/O devices.

FEATURES

- **Bidirectional 8-bit MicroController ($\overline{IV}$) bus**
- **User bus—four input bits and four output bits**
- **Independent bus operation**
- **Synchronous user data input**
- **Programmed MicroController port address**
- **Three-state TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 or 8X300 MicroControllers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

8X382 PACKAGE and PIN DESIGNATIONS



PIN NO.	IDENTIFIER	FUNCTION
1-4	UD7-UD4	Three-state, dedicated output lines for user data. UD7 corresponds to $\overline{IV7}$
5-8	UD3-UD0	Dedicated input lines for user data. UD0 corresponds to $\overline{IV0}$
9	$\overline{UOC}$	User Output Control—active low input to enable data output to UD4-UD7
10	$\overline{UIC}$	User Input Control—active low input to enable data input to UD0-UD3
11	$\overline{ME}$	Master Enable—active low input to enable the $\overline{IV}$ bus for data input, data output, or $\overline{IV}$ address selection/deselection. UD-bus operations are unaffected
12	GND	Ground
13	MCLK	Master Clock—active high input from MicroController used to strobe data into data latches from the $\overline{IV}$ bus and bits UD0-UD3 of the UD bus. MCLK also synchronizes $\overline{IV}$ address selection
14	SC	Select Command—active high input from MicroController to enable $\overline{IV}$ address input from the $\overline{IV}$ bus for device selection
15	WC	Write Command—active high from MicroController to enable the writing of data into the data latches from the $\overline{IV}$ bus, provided $\overline{UIC}$ is not low
16-23	$\overline{IV0}$-$\overline{IV7}$	Interface Vector Input/Output Bus—three-state bidirectional data bus. $\overline{IV0}$ corresponds to UD0
24	V.	Supply Voltage

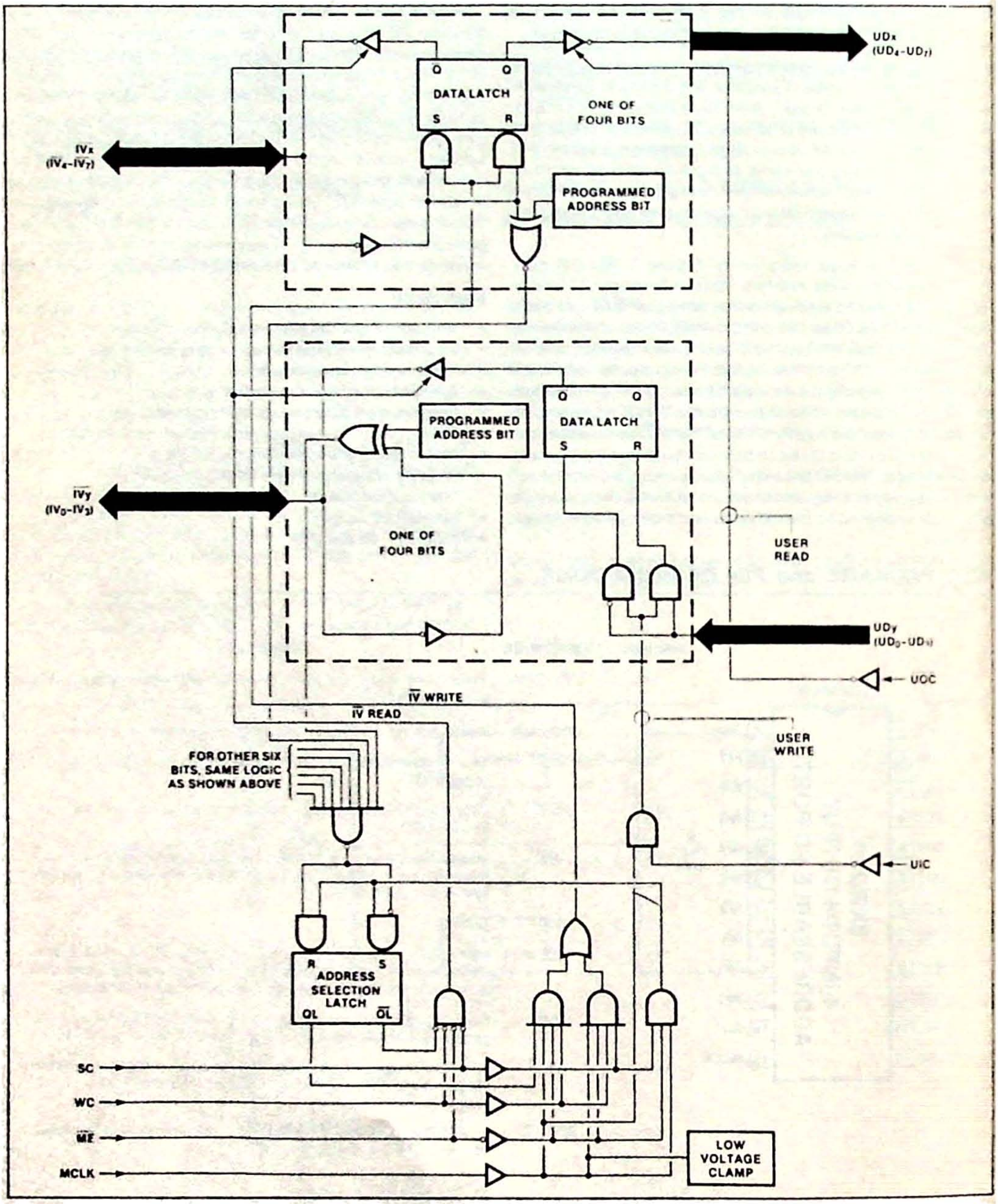

UDx
(UD4-UD7)
Q
Q
DATA LATCH
S
R
ONE OF
FOUR BITS
IVx
(IV4-IV7)
PROGRAMMED
ADDRESS BIT
PROGRAMMED
ADDRESS BIT
DATA LATCH
Q
Q
S
R
IVy
(IV0-IV3)
ONE OF
FOUR BITS
USER
READ
UDy
(UD0-UD3)
UOC
IV WRITE
IV READ
USER
WRITE
FOR OTHER SIX
BITS, SAME LOGIC
AS SHOWN ABOVE
UIC
R
S
ADDRESS
SELECTION
LATCH
QL
QL
SC
WC
ME
MCLK
LOW
VOLTAGE
CLAMP

Figure 1. Logic Diagram for 8X382 I/O Port

4-INPUT/4-OUTPUT ADDRESSABLE I/O PORT 8X382

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data-bus interface is controlled by the $\overline{UIC}$ and $\overline{UOC}$ inputs. Data input to UD0-UD3 is synchronous with MCLK, that is, with $\overline{UIC}$ low. Information is written into the data latches only when MCLK is high. The output drivers of UD4-UD7 bus are enabled when $\overline{UOC}$ is low.

Table 1. Input/Output Control of UD Bus

$\overline{UIC}$	$\overline{UOC}$	MCLK	FUNCTION OF UD BUS	
			UD0-UD3	UD4-UD7
H	L	X	Inactive	Output Data
L	X	H	Input Data	Inactive
L	X	L	Inactive	Inactive
H	H	X	Inactive	Inactive

X = don't care

$\overline{IV}$ Bus Control

Input/output control of the $\overline{IV}$ bus is shown in Table 2, this bus is controlled by SC, WC, $\overline{ME}$, MCLK and the current state of an internal address selection latch. The address selection latch in the I/O port stores the result of the most recent $\overline{IV}$ address selection. The latch is set when the internally preprogrammed address of the port matches the address on the $\overline{IV}$ bus during an address-selection operation (SC=MCLK=High/WC=Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The $\overline{IV}$ bus can transfer data only when the selection latch is set. The MicroController Left Bank ($\overline{LB}$) and Right Bank ($\overline{RB}$) outputs can control the $\overline{ME}$ inputs for two banks of I/O devices, thus, acting as ninth address bit.

Data is written into the data latches of a selected device from the $\overline{IV}$ bus when WC = MCLK = High and $\overline{ME}$ = Low. Output drivers on the $\overline{IV}$ bus of the device with the address latch set are enabled with $\overline{ME}$, WC, and SC low. With SC and WC both high (shaded entry of Table 2), the bit pattern present on $\overline{IV0}$-$\overline{IV7}$ is interpreted as both input data ($\overline{IV4}$-$\overline{IV7}$ only) and $\overline{IV}$ address. The data in $\overline{IV4}$-$\overline{IV7}$ is latched in whether or not the I/O port has been previously selected. If the preprogrammed address of the I/O port matches the bit pattern on $\overline{IV0}$-$\overline{IV7}$ when SC and WC are both high, the selection latch is set; otherwise, it is reset. (Note. *The MicroController never drives both SC and WC high at the same time.*)

Table 2. Input/Output Control of IV Bus

$\overline{ME}$	SC	WC	MCLK	SEL LATCH	FUNCTION OF $\overline{IV}$ BUS
L	L	L	X	Set	Output Data
L	L	H	H	Set	Input Data ($\overline{IV4}$-$\overline{IV7}$ only)
L	H	L	H	X	Input Address*
L	H	H	H	X	Input Data ($\overline{IV4}$-$\overline{IV7}$ only) and address*
L	X	H	L	X	Inactive
L	H	X	L	X	Inactive
L	L	X	X	Not set	Inactive
H	X	X	X	X	Inactive

X = don't care * Selection latch is updated

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note: *A logic "1" in MicroController software corresponds to a high level on the UD bus even though the $\overline{IV}$ bus is inverted.*) The 8X382 wakes up in the unselected state with all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

The 8X382 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level ($\leq$0.8V) inputs on all $\overline{IV}$ bus lines (address 255_{10}) will select the device. To program a given address bit to match a high level ($\geq$2.0V) input on the corresponding $\overline{IV}$ pin (a logical "0" to the MicroController), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state—$\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = +5V and SC = WC = MCLK = GND = 0.0V, leave the UD and $\overline{IV}$ bus pins open.

Step 2: Increase V_{CC} to 9.0 volts.

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level $\overline{IV}$ address bit. The I/O port is programmed from the user bus (UD0-UD7) for addressing from the MicroController bus ($\overline{IV0}$-$\overline{IV7}$).

Table 3. Programming Specifications

PARAMETERS	LIMITS			UNITS
	Min	Typ	Max	
V_{CCP} — Programming supply voltage Address	8.75	9.0	9.25	V
Protect		0		V
Maximum time V_{CCP} >5.25V			1.0	Sec
Programming voltage Address	8.75	9.0	9.25	V
Protect	13.5		14.0	V
Programming current Address			5	mA
Protect			75	mA
t_r — Programming pulse rise time Address	0.1		1.0	μS
Protect	0.1			μS
t_w — Programming pulse width	0.5		1.0	μS

Step 4: Return V_{CC} to 0-volts. (Note. *If the programming of all address bits is completed in less than 1-second, V_{CC} can remain at 9.0-volts for the required interval of time.*)

Step 5: Steps 1 through 3 are applicable to the programming of each address bit that requires a high-level $\overline{IV}$ match

Step 6: To verify that the address is properly programmed, return V_{CC} to +5V, set $\overline{IV0}$-$\overline{IV7}$ to the desired inverted binary address pattern, set $\overline{ME}$ = WC = Low and SC = MCLK = High. If there are no programming errors, subsequent data written from $\overline{IV4}$-$\overline{IV7}$ WC = High will appear inverted on UD4-UD7

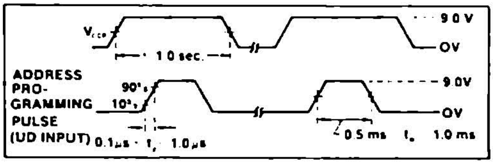


Figure 2. Address Programming Pulse

Address Protect

After programming the I/O Port, steps should be taken to isolate the address circuits and make these circuits permanently immune to further change.

Step 1: Set V_{CC} and all control inputs to 0-volts: V_{CC} = $\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = SC = WC = MCLK = GND = 0.0V; $\overline{IV0}$-$\overline{IV7}$ = open circuit

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0-UD7)—refer to Table 3 for min/max specifications pertaining to voltage and current

Step 3: Verify that the address circuits for each bit is isolated by applying 7-volts, in turn, to each user-bus pin (UD0-UD7) and measuring less than 1-milliampere of input current (Note: *Set-up conditions are the same as those in Step 1, the rise time of verification voltage must be 100-microseconds or slower*

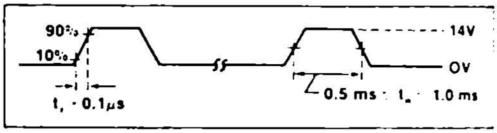


Figure 3. Protect Programming Pulse

DC ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage[1]	+7	V
V_{IN}	Input voltage[1]	+5.5	V
T_{STG}	Storage temperature range	-65 to +150	

COMMERCIAL: 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
MILITARY: 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_C ≤ 125°C

PARAMETER		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min, I_I = -10mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max, V_{IH} = 2.7V		5.0	100		5.0	100	µA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max, V_{IL} = 0.5V		-350	-550		-350	-550	µA
I_{OZH}	High-Z State Output Current—High Level[4]	V_{CC} = Max, V_{OH} = 2.5V			100			100	µA
I_{OZL}	High-Z State Output Current—Low Level[4]	V_{CC} = Max, V_{OL} = 0.5V			-100			-100	µA
V_{OL}	Low Level Output Voltage $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Min, I_{OL} = 16mA			0.55			0.55	V
	User Bus (UD4-UD7)	V_{CC} = Min, I_{OL} = 24mA			0.55			0.55	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = -3.2mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[2] $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD4-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max, $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150		90	150	mA

Notes
[1] The input current includes the Three-state leakage current of the output driver on the data lines
[2] Only one output may be shorted at a time
[3] These limits do not apply during address programming
[4] Applies only to pins UD4-UD7

4-INPUT/4-OUTPUT ADDRESSABLE I/O PORT

8X382

AC ELECTRICAL CHARACTERISTICS

COMMERCIAL 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
MILITARY 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_C ≤ 125°C

LOADING: See TEST LOADING CIRCUITS

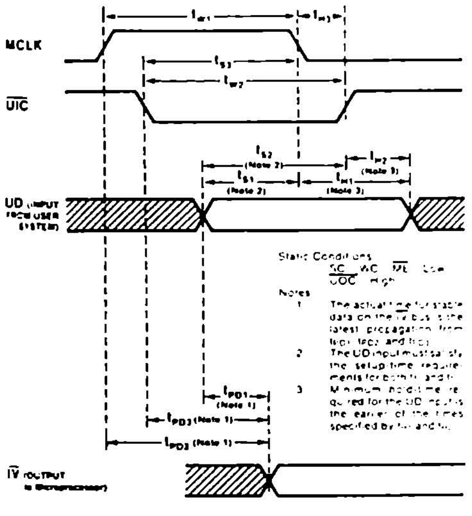
PARAMETER		REFERENCES[1]		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
		FROM	TO		Min	Typ	Max	Min	Typ	Max	
Pulse Widths:											
t_{W1}	Clock High	↑MCLK	↓MCLK		35			35			ns
t_{W2}	User Input Control	↓$\overline{UIC}$	↑$\overline{UIC}$	MCLK = High	35			35			ns
Propagation Delays:											
t_{PD1}	UD Propagation Delay	UD_{0-3}	$\overline{IV}_{0-3}$	MCLK = High SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			30			30	ns
t_{PD2}	UD Clock Delay	↑MCLK	$\overline{IV}_{0-3}$	UD_{0-3} = Stable, MCLK = High SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50			50	ns
t_{PD3}	UD Input Delay	↓$\overline{UIC}$	$\overline{IV}_{0-3}$	UD_{0-3} = Stable, MCLK = High SC = WC = $\overline{ME}$ = Low			50			50	ns
t_{PD4}	$\overline{IV}$ Data Propagation Delay	$\overline{IV}_{4-7}$	UD_{4-7}	MCLK = WC = High $\overline{ME}$ = $\overline{UOC}$ = SC = Low			45			45	ns
t_{PD5}	$\overline{IV}$ Data Clock Delay	↑MCLK	UD_{4-7}	$\overline{IV}_{4-7}$ = Stable, WC = High $\overline{ME}$ = $\overline{UOC}$ = SC = Low			55			55	ns
Output Enable Timing:											
t_{OE1}	UD Output Enable	↓$\overline{UOC}$	UD_{4-7}				30			30	ns
t_{OE2}	$\overline{IV}$ Data Master Enable	↓$\overline{ME}$	$\overline{IV}$	WC = SC = Low			22			25	ns
t_{OE3}	$\overline{IV}$ Data Write Recovery	↓WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25			25	ns
t_{OE4}	$\overline{IV}$ Data Select Recovery	↓SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			25			25	ns
Output Disable Timing:											
t_{OD1}	UD Output Disable	↑$\overline{UOC}$	UD_{4-7}				25			25	ns
t_{OD2}	$\overline{IV}$ Data Master Disable	↑$\overline{ME}$	$\overline{IV}$	WC = SC = Low			25			25	ns
t_{OD3}	$\overline{IV}$ Data Write Override	↑WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			20			20	ns
t_{OD4}	$\overline{IV}$ Data Select Override	↑SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			20			20	ns
Setup Times:											
t_{S1}	UD Clock Setup Time	UD_{0-3}	↓MCLK	$\overline{UIC}$ = Low	15			15			ns
t_{S2}	UD Control Setup Time	UD_{0-3}	↑$\overline{UIC}$	MCLK = High	15			15			ns
t_{S3}	User Input Control Setup Time	↓$\overline{UIC}$	↓MCLK		25			25			ns
t_{S4}	$\overline{IV}$ Data Setup Time	$\overline{IV}$	↓MCLK	WC = High or SC = High, $\overline{ME}$ = Low.	35			35			ns
t_{S5}	$\overline{IV}$ Master Enable Setup Time	↓$\overline{ME}$	↓MCLK	WC = High or SC = High	30			30			ns
t_{S6}	$\overline{IV}$ Write Control Setup Time	↑WC	↓MCLK	SC = $\overline{ME}$ = Low.	30			30			ns
t_{S7}	$\overline{IV}$ Select Control Setup Time	↑SC	↓MCLK	WC = $\overline{ME}$ = Low	30			30			ns

AC ELECTRICAL CHARACTERISTICS (Cont'd)

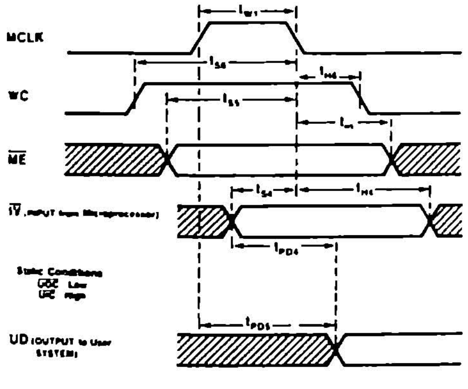
PARAMETER		REFERENCES		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
		FROM	TO		Min	Typ	Max	Min	Typ	Max	
Hold Times:											
t_{H1}	UD Clock Hold Time	↓MCLK	UD	$\overline{\text{UIC}}$ = Low	15			15			ns
t_{H2}	UD Control Hold Time	↑$\overline{\text{UIC}}$	UD	MCLK = High	15			15			ns
t_{H3}	User Input Control Hold Time	↓MCLK	↑$\overline{\text{UIC}}$		0			0			ns
t_{H4}	Data Hold Time	↓MCLK	$\overline{\text{IV}}$	WC = High or SC = High, $\overline{\text{ME}}$ = Low	5			5			ns
t_{H5}[3]	Master Enable Hold Time	↓MCLK	↑$\overline{\text{ME}}$	WC = High or SC = High	0			0			ns
t_{H6}	$\overline{\text{IV}}$ Write Control Hold Time	↑MCLK	↓WC	SC = $\overline{\text{ME}}$ = Low	0			0			ns
t_{H7}	$\overline{\text{IV}}$ Select Control Hold Time	↑MCLK	↓SC	WC = $\overline{\text{ME}}$ = Low	0			0			ns

Notes

1. All measurements to the $\overline{\text{IV}}$ bus assumes the address select on latch is set.
2. These parameters are measured with a capacitive loading of 50 pF and represent the output driver turn off time.
3. If $\overline{\text{ME}}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended write or [illegible] selection of the I/O port.


MCLK
$\overline{UIC}$
UD (INPUT FROM USER SYSTEM)
$\overline{IV}$ (OUTPUT to Microprocessor)
t_{W1}
t_{H3}
t_{S3}
t_{W2}
t_{S2} (Note 2)
t_{H2} (Note 3)
t_{S1} (Note 2)
t_{H1} (Note 3)
t_{PD1} (Note 1)
t_{PD3} (Note 1)
t_{PD2} (Note 1)
Static Conditions
SC WC $\overline{ME}$ Low
$\overline{UOC}$ High
Notes
1
2
3

a. User Data Input Timing

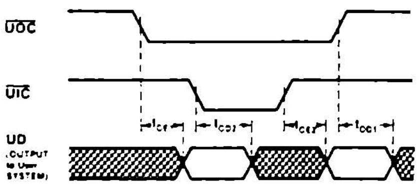

MCLK
WC
$\overline{ME}$
$\overline{IV}$ (INPUT from Microprocessor)
UD (OUTPUT to User SYSTEM)
t_{W1}
t_{S6}
t_{H6}
t_{S5}
t_{H5}
t_{S4}
t_{H4}
t_{PD4}
t_{PD5}
Static Conditions
$\overline{UOC}$ Low
$\overline{UIC}$ High

c. MicroController Write Cycle Timing

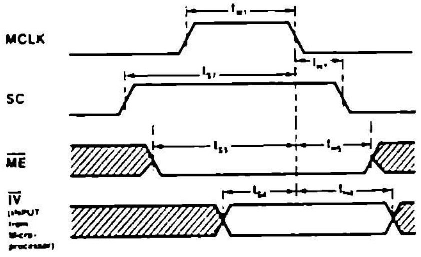
Legend:

▨ - THREE-STATE

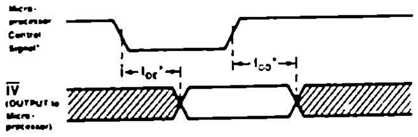
▨ - CHANGING DATA


$\overline{UOC}$
$\overline{UIC}$
UD (OUTPUT to User SYSTEM)
t_{CE1}
t_{CD2}
t_{CE2}
t_{CD1}

b. User Data Output Enable Timing


MCLK
SC
$\overline{ME}$
$\overline{IV}$ (INPUT from Micro-processor)
t_{W1}
t_{S7}
t_{H7}
t_{S3}
t_{H3}
t_{S4}
t_{H4}

d. MicroController Select Cycle Timing


Micro-processor Control Signal*
$\overline{IV}$ (OUTPUT to Micro-processor)
t_{OE}*
t_{OD}*

*PARAMETER KEY

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
$\overline{ME}$	t_{OE3} t_{OD3}	SC WC LOW
WC	t_{OE5} t_{OD5}	SC $\overline{ME}$ LOW
SC	t_{OE6} t_{OD6}	WC $\overline{ME}$ LOW

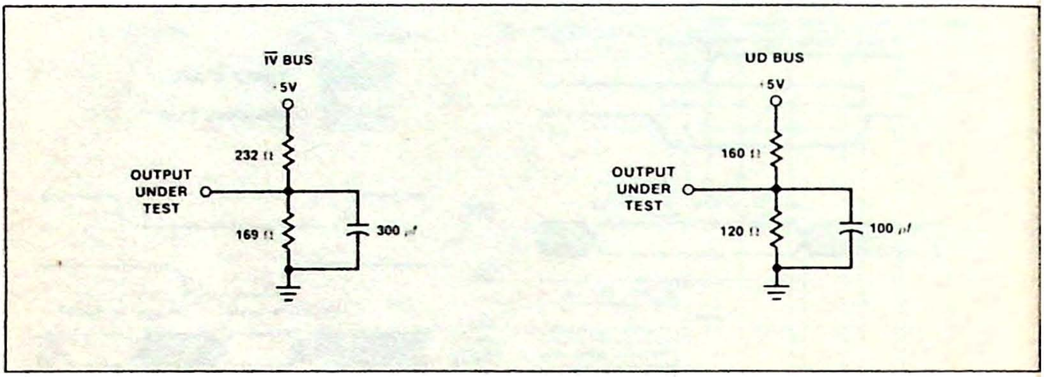
e. MicroController Output Enable Timing

Figure 4. Timing Diagrams

4-INPUT/4-OUTPUT ADDRESSABLE I/O PORT

8X382

TEST LOADING CIRCUITS



ORDERING INFORMATION

Preprogrammed Addresses

The 8X382 can be ordered with addresses preprogrammed at the factory. To order use the following part-number format

COMMERCIAL –
N8X382N-*zzz* Plastic
N8X382I-*zzz* Ceramic
MILITARY –
S8X382I/883B-*zzz*
S8X382I/883C-*zzz*

where. *zzz* any address from 000_{10} through 255_{10}. Note *I/O Ports with preprogrammed addresses of 0_{10} through 15_{10} are stock items. consult the nearest Signetics Sales and Service Office when ordering parts in the address range of 16_{10} through 255_{10}.*

Field-Programmable Addresses

Relevant ordering information for these parts is given below; unless otherwise indicated, referenced packaging data and operating specifications for parts with preassigned or field-programmable addresses are the same

COMMERCIAL –
N8X382N Plastic
N8X382I Ceramic
MILITARY –
S8X382I/883B
S8X382I/883C

APPLICATIONS

When compared to other MicroController ports in the 8X370 series, the 8X382 has some unique features that provide real design advantages in certain applications. Connection of the I/O port to the MicroController is simple and straightforward in that like pin names are tied together. The system designer must also decide on which bank of the MicroController to place the 8X382 and then connect the $\overline{ME}$ pin of the port to either the $\overline{LB}$ (Left Bank) or $\overline{RB}$ (Right Bank) of the MicroController.

The 8X382 is unique because it can be used for both dedicated input and output operations. In the system shown below, the user interface requires nine (9) dedicated inputs and eleven (11) dedicated outputs. Observe that by using an 8X382, the problem is solved by three devices whereas four 8X372 ports are required for the same solution.

Another important use of the 8X382 is in implementing a handshake interface. Since both input and output bits reside in the same port, I/O operations can be performed without port re-addressing. Users may also find the 8X382 an advantage in the layout of Printed Circuit boards, since random control/status signals can be grouped within the same device position.

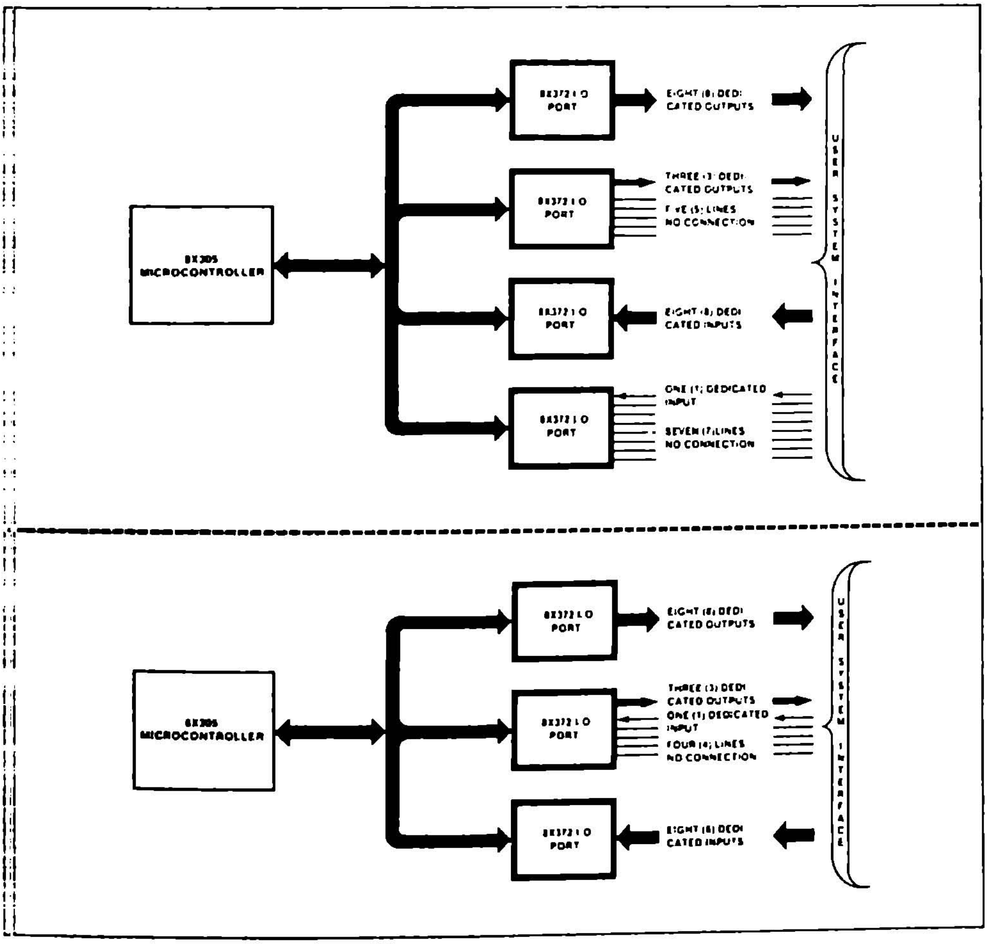
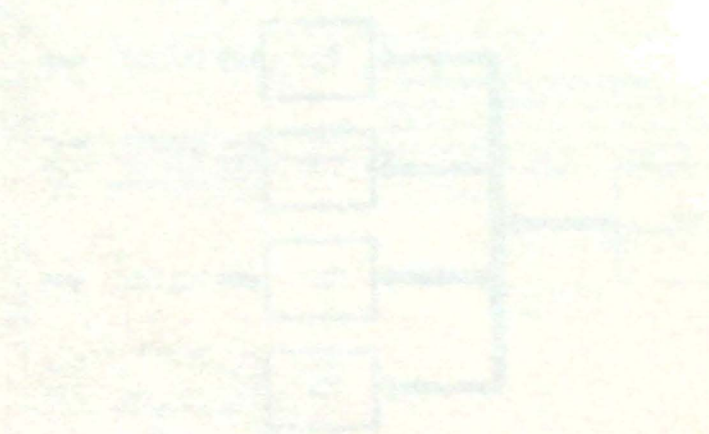
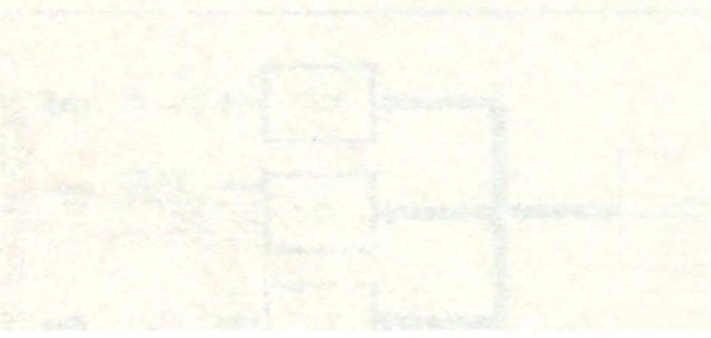

8X305 MICROCONTROLLER
8X372 I/O PORT
EIGHT (8) DEDICATED OUTPUTS
8X372 I/O PORT
THREE (3) DEDICATED OUTPUTS
FIVE (5) LINES NO CONNECTION
8X372 I/O PORT
EIGHT (8) DEDICATED INPUTS
8X372 I/O PORT
ONE (1) DEDICATED INPUT
SEVEN (7) LINES NO CONNECTION
USER SYSTEM INTERFACE
8X305 MICROCONTROLLER
8X372 I/O PORT
EIGHT (8) DEDICATED OUTPUTS
8X382 I/O PORT
THREE (3) DEDICATED OUTPUTS
ONE (1) DEDICATED INPUT
FOUR (4) LINES NO CONNECTION
8X372 I/O PORT
EIGHT (8) DEDICATED INPUTS
USER SYSTEM INTERFACE

Figure 5. Logic Diagram for 8X382 I/O Port

[illegible]

[illegible]





8-BIT LATCHED BIDIRECTIONAL I/O PORT

8T31/8X31

PRODUCT DESCRIPTION

Both the 8T31 and 8X31 are 8-bit bidirectional data registers designed to function as Input/Output interface elements in microprocessor systems. The two parts are functionally identical, however the 8X31 is housed in a 0.4-inch Slim-Line package which makes it ideally suited for applications where board space is limited.

Each part contains eight clocked data latches that are accessible from either a *microprocessor* port or a *user* port. Separate I/O control is provided for each port. The two ports operate independently except that when both are attempting to input data into the data latches, the User port (UD0-UD7) has priority. The master enable ($\overline{ME}$) signal enables or disables the microprocessor bus regardless of the state of the other inputs but has no effect on the user bus.

A unique feature of these parts is their ability to start up in a predetermined state. If the clock is maintained at a level of less than 0.8 volts until the power supply reaches 3.5 volts, all bits of the user port will wakeup at a "logic 1" level and those of the microprocessor port will wakeup in the high-impedance state.

FEATURES

- Dual bidirectional ports
- Independent port operation (User-port priority for data entry)
- User data input synchronous
- At power-up, User-port outputs are high and Microprocessor-port outputs are high-Z
- Three-state TTL outputs for high-drive capabilities
- Directly compatible with 8X300 Microcontroller
- Single +5V supply
- Slim (0.4 in.) 24-pin DIP (8X31 only)

ORDERING INFORMATION

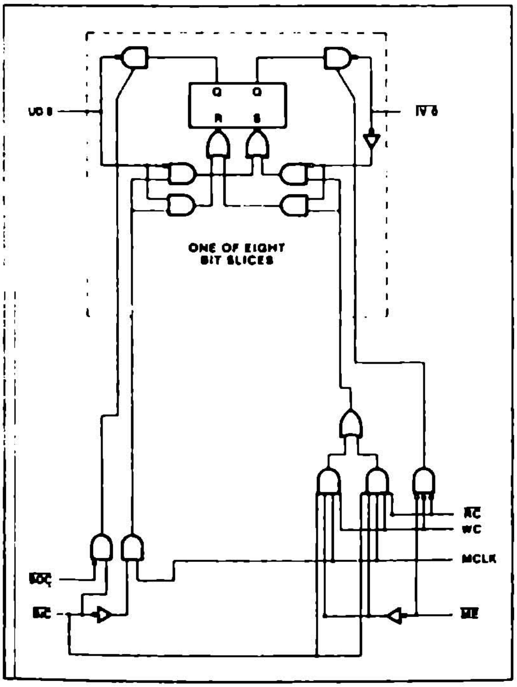
Order number: N8T31N, N8X31N

Packaging information: Refer to Signetics price list

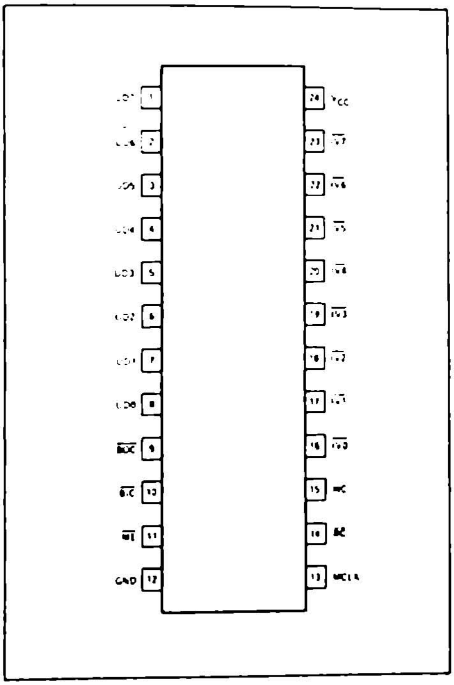
Supply voltage: 5V (±5%)

Operating temperature range: 0°C to +70°C

BLOCK DIAGRAM



PIN CONFIGURATION



PIN DESIGNATION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
1-8	UD0-UD7	User Data I/O Lines Bidirectional data lines to communicate with user's equipment	Acitve high three-state
16-23	$\overline{IV0}$-$\overline{IV7}$	Microprocessor Bus Bidirectional data lines to communicate with controlling digital system	Active low three-state
10	$\overline{BIC}$	Input Control User input to control writing into the I/O Port from the user data lines	Active low
9	$\overline{BOC}$	Output Control User input to control reading from the I/O Port onto the user data lines	Active low
11	$\overline{ME}$	Master Enable System input to enable or disable all other system inputs and outputs It has no effect on user inputs and outputs	Active low
15	WC	Write Command When WC is high stores contents of $\overline{IV0}$-$\overline{IV7}$ as data	Active high
14	$\overline{RC}$	Read Command When RC is low, data is presented on $\overline{IV0}$-$\overline{IV7}$	Active low
13	MCLK	Master Clock Input to strobe data into the latches See function tables for details	Active high
24	V_{CC}	5V power connection	
12	GND	Ground	

$\overline{BIC}$	$\overline{BOC}$	MCLK	USER DATA BUS FUNCTION
H	L	X	Output Data
L	X	H	Input Data
H	H	X	Inactive

H High Level L Low Level X Don't care

Table 1 USER PORT CONTROL FUNCTION

$\overline{ME}$	$\overline{RC}$	WC	MCLK	$\overline{BIC}$	MICROPROCESSOR BUS FUNCTION
L	L	L	X	X	Output Data
L	X	H	H	H	Input Data
X	H	L	X	X	Inactive
X	X	H	X	L	Inactive
H	X	X	X	X	Inactive

Table 2 MICROPROCESSOR PORT CONTROL FUNCTION

USER DATA BUS CONTROL

The activity of the user data bus is controlled by the $\overline{BIC}$ and $\overline{BOC}$ inputs as shown in Table 1

The user data input is a synchronous function with MCLK A low level on the $\overline{BIC}$ input allows data on the user data bus to be written into the data latches only if MCLK is at a high level

To avoid conflicts at the data latches input from the microprocessor port is inhibited when $\overline{BIC}$ is at a low level Under all other conditions the two ports operate independently

MICROPROCESSOR BUS CONTROL

As is shown in Table 2, the activity of the microprocessor port is controlled by the $\overline{ME}$ $\overline{RC}$ WC and $\overline{BIC}$ inputs as well as the state of an internal status latch $\overline{BIC}$ is included to show user port priority over the microprocessor port for data input

BUS OPERATION

Data written into the 8T31 8X31 from one port will appear inverted when read from the other port Data written into the 8T31 8X31 from one port will not be inverted when read from the same port

DC ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ± 5%, 0°C ≤ T_A ≤ 70°C unless otherwise specified

PARAMETER		TEST CONDITIONS	LIMITS Min	Typ	Max	UNIT
Input voltage						V
V_{IH}	High		2.0			
V_{IL}	Low				8	
V_{IC}	Clamp	I_I = -5mA			-1	
Output voltage		V_{CC} = 4.75V				V
V_{OH}	High		2.4			
V_{OL}	Low				55	
Input current[1]		V_{CC} = 5.25V				µA
I_{IH}	High	V_{IH} = 5.25V		<10	100	
I_{IL}	Low	V_{IL} 5V		-350	-550	
Output current[2]						mA
I_{OS}	Short circuit	V_{CC} = 4.75V				
	UD bus		10			
	IV bus		20			
I_{CC}	VCC supply current	V_{CC} = 5.25V		100	150	mA

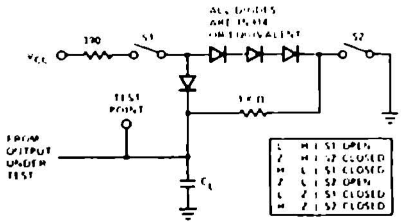
NOTES

1 The input current includes the three-state open collector leakage current of the output driver on the data lines

2 Only one output may be shorted at a time

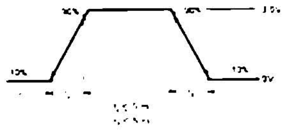
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT FOR THREE-STATE OUTPUTS

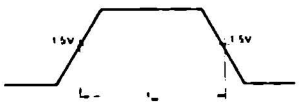

VCC
390
S1
ALL DIODES ARE 1N914 OR EQUIVALENT
S2
TEST POINT
1 K Ω
FROM OUTPUT UNDER TEST
CL
L H S1 OPEN
Z H S2 CLOSED
H L S1 CLOSED
Z L S2 OPEN
L Z S1 CLOSED
H Z S2 CLOSED

NOTE C_L includes fixture capacitance

INPUT WAVEFORM

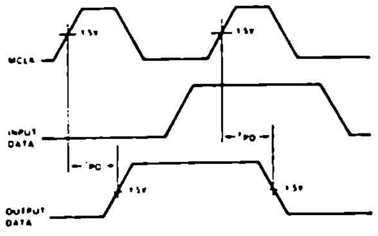

90%
90%
3.0V
10%
10%
0V

CLOCK PULSE WIDTH



DATA DELAY TIMES

Clock Referenced


MCLK
1.5V
1.5V
INPUT DATA
tPD
tPD
OUTPUT DATA
1.5V
1.5V

DC ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ± 5%, 0°C ≤ T_A ≤ 70°C unless otherwise specified

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Input voltage						V
V_{IH}	High		2 0			
V_{IL}	Low				8	
V_{IC}	Clamp	I_I = -5mA			-1	
Output voltage		V_{CC} = 4 75V				V
V_{OH}	High		2 4			
V_{OL}	Low				55	
Input current[1]		V_{CC} = 5 25V				µA
I_{IH}	High	V_{IH} = 5 25V		<10	100	
I_{IL}	Low	V_{IL} = 5V		-350	-550	
Output current[2]						mA
I_{OS}	Short circuit	V_{CC} = 4 75V				
	UD bus		10			
	IV bus		20			
I_{CC}	VCC supply current	V_{CC} = 5 25V		100	150	mA

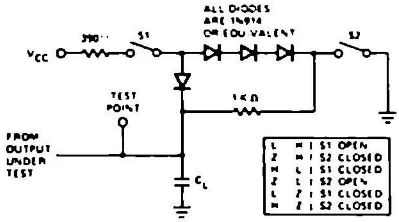
NOTES

1 The input current includes the three-state open collector leakage current of the output driver on the data lines

2 Only one output may be shorted at a time

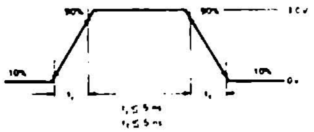
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT FOR THREE-STATE OUTPUTS

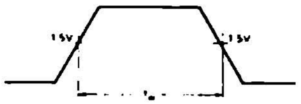

ALL DIODES ARE 1N914 OR EQUIVALENT
390Ω
S1
S2
VCC
1KΩ
TEST POINT
FROM OUTPUT UNDER TEST
CL
L H 1 S1 OPEN
Z H 1 S2 CLOSED
H L 1 S1 CLOSED
Z L 1 S2 OPEN
L Z 1 S1 CLOSED
H Z 1 S2 CLOSED

NOTE C_L includes fixture capacitance

INPUT WAVEFORM

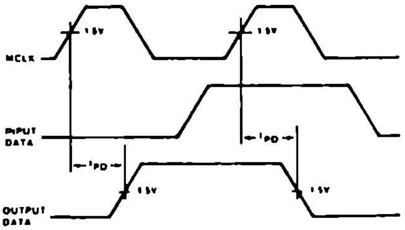

90%
90%
3.0V
10%
10%
0V
tr
tf
tr ≤ 5ns
tf ≤ 5ns

CLOCK PULSE WIDTH



DATA DELAY TIMES

Clock Referenced


1 5V
1 5V
MCLK
INPUT DATA
tPD
tPD
1 5V
1 5V
OUTPUT DATA

8-BIT LATCHED ADDRESSABLE BIDIRECTIONAL I/O PORT

8T32/8T33/8T35/8T36 8X32/8X36/8X42

PRODUCT IDENTITY

8T32—	Three-state, field-programmable (addresses 0-255), synchronous user port
8X32—	Three-state, preprogrammed addresses (0-15), synchronous user port in slim-line (0.4 in.) package
8T33—	Open-collector, synchronous user port
8T35—	Open-collector, asynchronous user port
8T36—	Three-state, field-programmable (addresses 0-255), asynchronous user port
8X36—	Three-state, preprogrammed addresses (0-15), asynchronous user port in slim-line (0.4 in.) package
8X42—	Three-state, 4-input/4-output, preprogrammed addresses (0-15), synchronous user port

ORDERING INFORMATION

All of the above parts can be ordered as field-programmable or with addresses preassigned at the factory. To order, use the following part-number format

N8XYY-ZZZ-P where
N8TYY-ZZZ-P

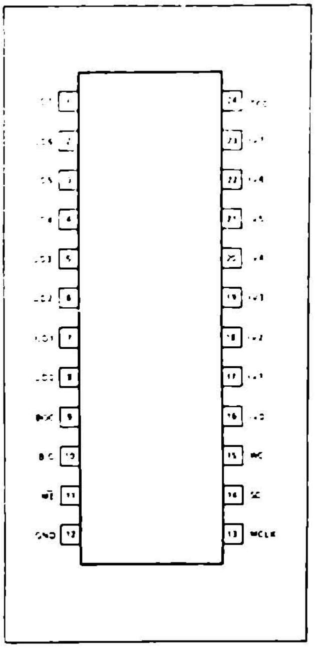
P = F for Ceramic package and
NA for Plastic package
ZZZ = Any address from 000 through 255 (decimal)—256 available addresses for each 8T-- part. For an 8X part, preassigned addresses are limited to 000 through 015 (decimal)—16 available addresses
YY = I/O port version

FEATURES

- Dual bidirectional ports (except 8X42)
- Independent port operation (user-port priority for data entry)
- User data input available as synchronous (8T32/8X32 8X42/8T33) or as asynchronous (8T35/8T36/8X36)
- User data bus available with three-state (8T32/8X32, 8T36/8X36, and/or 8X42) or open-collector (8T33 and/or 8T35) outputs
- At power-up, user-port outputs are high and microprocessor-port outputs are high-z; status latch (from address compare) is also cleared at power-up
- Three-state TTL outputs for high-drive capabilities
- Directly compatible with 8X300 microcontroller
- Single +5V supply
- Slim (0.4 in.) 24-pin DIP (8X32/8X36/8X42 only)

A stock of 8T32s and 8T35s with addresses "1" through "10" are maintained in inventory with a longer lead time, a small quantity of addresses "11" through "50" are also available

PIN CONFIGURATION



PRODUCT DESCRIPTIONS

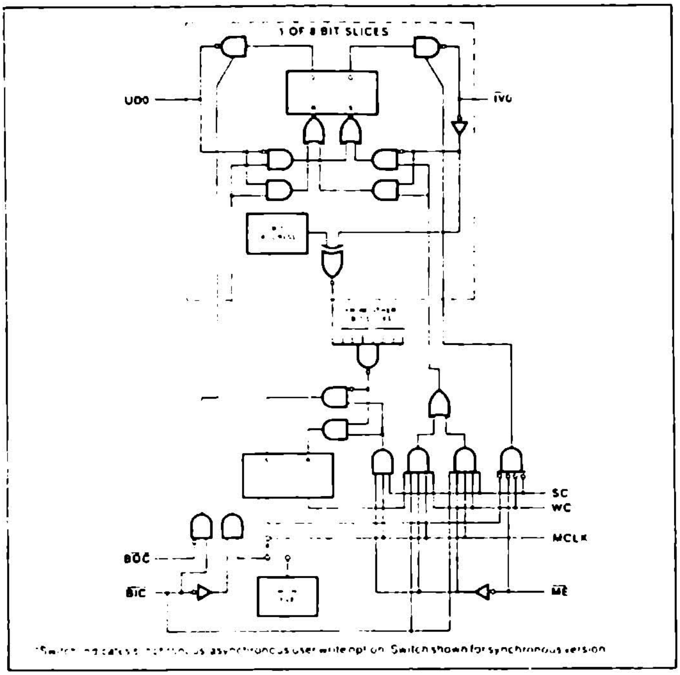
8T32/8T33/8T35/8T36. Each of these I/O Bytes is an addressable and bi-directional register designed for use as an interface element in any system with TTL-compatible buses. (*Note.* Since these I/O Bytes are frequently used with the 8X300 Microcontroller and its associated Interface Vector bus, the 8T32-8T36 family of parts are commonly called IV Bytes.) Each I/O Byte contains eight identical data latches (Bits 0 through 7), the latches are accessed from either of two 8-bit ports—one port connecting to the microprocessor (8X300) and the other port connecting to the user device. Separate controls are provided for each port and the two ports operate independently, except when both attempt to input data at the same time, in this case, the user port bus has priority.

The address of each I/O Byte is field-programmable and the microprocessor port is accessed when a valid address is received, the user port is accessible at all times. A selected Byte is automatically deselected when the address of another I/O Byte is sensed on the address/data bus. A Master Enable ($\overline{ME}$) input is available for use as a ninth address bit, allowing direct access to 512 I/O Bytes without address decoding.

A unique feature of these parts is their ability to start up in a predetermined state. If the clock is maintained at a level of less than 0.8 volts until the power supply reaches 3.5 volts, all bits of the user port will wakeup at a "logic 1" level and those of the microprocessor port will wakeup in the high-impedance state.

8X32/8X36. Except for package differences and the fact that the 8X32 and 8X36 are available with preprogrammed addresses (0

TYPICAL BLOCK DIAGRAM


1 OF 8 BIT SLICES
UDO
IVO
SC
WC
MCLK
$\overline{BOC}$
$\overline{BIC}$
$\overline{ME}$

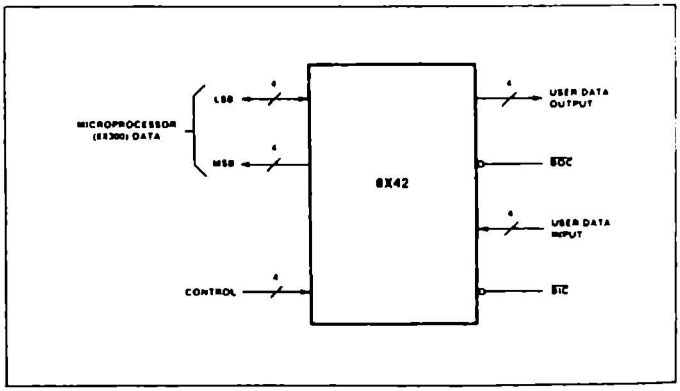
*Switch indicates synchronous/asynchronous user write option. Switch shown for synchronous version.

through 15), these two devices are functionally identical to their 8Txx counterparts; design parameters and interface requirements are exactly the same.

8X42. This part differs from the 8X32 in the following ways.

- Bits 0-3 (pins 8-5, respectively) are dedicated user inputs controlled by Byte Input Control ($\overline{BIC}$/pin 10). Byte Output Control ($\overline{BOC}$/pin 9) does not affect these bits and the user output drivers do not exist for these bits.
- Bits 4-7 (pins 4-1, respectively) are dedicated user outputs whose three-state drivers are controlled by $\overline{BOC}$. $\overline{BIC}$ has no control over these bits.
- Pin numbers, software requirements, and AC characteristics of the 8X42 are identical to those of the 8X32, ergo 8T32. A simplified block diagram of the 8X42 is shown in the accompanying diagram.

SIMPLIFIED DIAGRAM OF 8X42


MICROPROCESSOR (8X300) DATA
LSB
MSB
CONTROL
4
4
4
4
4
8X42
USER DATA OUTPUT
$\overline{BOC}$
USER DATA INPUT
$\overline{BIC}$

PIN DESCRIPTION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
1-8	UD0-UD7	User Data I/O Lines Bidirectional data lines to communicate with user's equipment Either tri-state or open collector outputs are available	Active high
16-23	$\overline{IV0}$-$\overline{IV7}$	Microprocessor Bus Bidirectional data lines to communicate with controlling digital system (microprocessor)	Active low three-state
10	$\overline{BIC}$	Input Control User input to control writing into the I/O Port from the user data lines	Active low
9	$\overline{BOC}$	Output Control User input to control reading from the I/O Port onto the user data lines	Active low
11	$\overline{ME}$	Master Enable System input to enable or disable all other system inputs and outputs It has no effect on user inputs and outputs	Active low
15	WC	Write Command When WC is high and SC is low, I/O Port, if selected, stores contents of $\overline{IV0}$-$\overline{IV7}$ as data	Active high
14	SC	Select Command When SC is high and WC is low, data on $\overline{IV0}$-$\overline{IV7}$ is interpreted as an address I/O Port selects itself if its address is identical to μP bus data, it de-selects itself otherwise	Active high
13	MCLK	Master Clock Input to strobe data into the latches See function tables for details	Active high
24	VCC	5V power connection	
12	GND	Ground	

$\overline{BIC}$	$\overline{BOC}$	MCLK	USER DATA BUS FUNCTION	
			8T32/8X32/8T33	8T35/8T36/8X36
H	L	X	Output Data	Output Data
L	X	H	Input Data	Input Data
L	X	L	Inactive	Input Data
H	H	X	Inactive	Inactive

H = High Level L = Low Level X = Don't care

Table 1 USER PORT CONTROL FUNCTION

$\overline{ME}$	SC	WC	MCLK	$\overline{BIC}$	STATUS LATCH	I/O PORT FUNCTION
L	L	L	X	X	SET	Output Data
L	L	H	H	H	SET	Input Data
L	H	L	H	X	X	Input Address
L	H	H	H	L	X	Input Address
L	H	H	H	H	X	Input Data and Address
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

Table 2 MICROPROCESSOR PORT CONTROL FUNCTION

USER DATA BUS CONTROL

The activity of the user data bus is controlled by the $\overline{BIC}$ and $\overline{BOC}$ inputs as shown in Table 1

For the 8T32 8X32, 8T33, and 8X42, user data input is a synchronous function with MCLK A low level on the $\overline{BIC}$ input allows data on the user data bus to be written into the data latches only if MCLK is at a high level For the 8T35, 8T36 and 8X36, user data input is an asynchronous function A low level on the $\overline{BIC}$ input allows data on the user data bus to be latched regardless of the level of the MCLK input Note that when the 8T35, 8T36, or 8X36 is used with the 8X300 Microcontroller, care must be taken to insure that the Microprocessor bus is stable when it is being read by the 8X300 Microcontroller

To avoid conflicts at the Data Latches, input from the Microprocessor Port is inhibited when $\overline{BIC}$ is at a low level Under all other conditions the two ports operate independently

MICROPROCESSOR BUS CONTROL

As is shown in Table 2, the activity of the microprocessor port is controlled by the $\overline{ME}$, SC, WC and $\overline{BIC}$ inputs, as well as the state of an internal status latch $\overline{BIC}$ is included to show user port priority over the microprocessor port for data input

Each I/O Port's status latch stores the result of the most recent I/O Port select it is set when the I/O Port's internal address matches the Microprocessor Bus It is cleared when an address that differs from the internal address is presented on the Microprocessor Bus

In normal operation, the state of the status latch acts like a master enable, the microprocessor port can transfer data only when the status latch is set

When SC and WC are both high, data on the Microprocessor Bus is accepted as data, whether or not the I/O Port was selected The data is also interpreted as an address The I/O Port sets its select status if its address matches the data read when SC and WC were both high; it resets its select status otherwise

BUS OPERATION

Data written into the I/O Port from one port will appear inverted when read from the other port Data written into the I/O Port from one port will not be inverted when read from the same port

AC ELECTRICAL CHARACTERISTICS $0°C \le T_A \le 70°C$, $V_{CC} = 5V \pm 5\%$

PARAMETER		INPUT	TEST CONDITION	LIMITS			UNIT
				Min	Typ	Max	
t_{PD}	User data delay (Note 1)	UD X	$C_L = 50pF$		25	38	ns
		MCLK*			45	61	
		$\overline{\text{BIC}}$†			40	55	
t_{OE}	User output enable	$\overline{\text{BOC}}$	$C_L = 50pF$	18	26	47	ns
t_{OD}	User output disable	$\overline{\text{BIC}}$	$C_L = 50pF$	18	28	35	ns
		$\overline{\text{BOC}}$		16	23	33	
t_{PD}	µP data delay (Note 1)	$\overline{\text{IV}}$ X	$C_L = 50pF$		38	53	ns
		MCLK			48	61	
t_{OE}	µP output enable	$\overline{\text{ME}}$ SC WC	$C_L = 50pF$	14	19	25	ns
t_{OD}	µP output disable	$\overline{\text{ME}}$ SC WC	$C_L = 50pF$	13	17	32	ns
t_W	Minimum pulse width	MCLK		40			ns
		$\overline{\text{BIC}}$†		35			
t_{SETUP}	Minimum setup time	UD X□	(Note 2)	15			ns
		$\overline{\text{BIC}}$*		25			
		$\overline{\text{IV}}$ X		55			
		$\overline{\text{ME}}$		30			
		SC		30			
		WC		30			
t_{HOLD}	Minimum hold time	UD X□	(Note 2)	25			ns
		$\overline{\text{BIC}}$*		10			
		$\overline{\text{IV}}$ X		10			
		$\overline{\text{ME}}$		5			
		SC		5			
		WC		5			

* Applies for 8T32, 8X32 and 8T33 only

† Applies for 8T35, 8T36 and 8X36 only

□ Times are referenced to MCLK for 8T32, 8X32 and 8T33, and are referenced to $\overline{\text{BIC}}$ for 8T35, 8T36 and 8X36

NOTES

1. Data delays referenced to the clock are valid only if the input data is stable at the arrival of the clock and the hold time requirement is met.
2. Set up and hold times given are for "normal" operation. $\overline{\text{BIC}}$ setup and hold times are for a user write operation. SC setup and hold times are for an I/O Port select operation. WC setup and hold times are for a Microprocessor Bus write operation. $\overline{\text{ME}}$ setup and hold times are for both IV write and select operations.

DC ELECTRICAL CHARACTERISTICS $0°C \le T_A \le 70°C, V_{CC} = 5V \pm 5\%$

PARAMETER		TEST CONDITIONS	LIMITS			UNITS
			Min	Typ	Max	
V_{IH}	High-level input voltage		2.0		5.5	V
V_{IL}	Low-level input voltage		-1.0		8	V
V_{CL}	Input clamp voltage	$I_I = -5mA$			-1.0	V
I_{IH}	High-level input current[1]	$V_{CC} = 5.25V$ $V_{IH} = 5.25V$		<10	100	µA
I_{IL}	Low level input current[1]	$V_{CC} = 5.25V$ $V_{IL} = .5V$		-350	-550	µA
V_{OL}	Low-level output voltage	$V_{CC} = 4.75V$ $I_{OL} = 16mA$			55	V
V_{OH}	High-level output voltage	$V_{CC} = 4.75V$ $I_{OH} = -3.2mA$	2.4			V
I_{OS}	Short-circuit output current[2]					
	UD bus	$V_{CC} = 4.75V$	10			mA
	IV bus	$V_{CC} = 4.75V$	20			mA
I_{CC}	Supply current	$V_{CC} = 5.25V$		100	150	mA

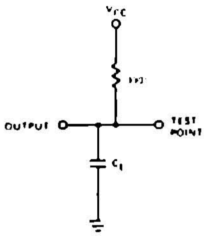
NOTES

1 The input current includes the Three-state Open Collector leakage current of the output driver on the data lines
2 Only one output may be shorted at a time
3 These limits do not apply during address programming

Absolute Maximum Ratings:

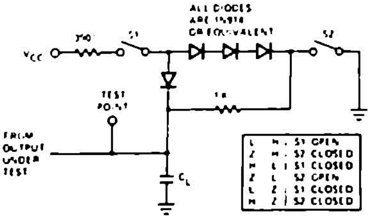
Supply voltage[3]	7V
Input voltage[3]	5.5V

TEST LOAD CIRCUIT (OPEN COLLECTOR OUTPUTS)


VCC
OUTPUT
TEST POINT
CL

NOTE C_L includes fixture capacitance

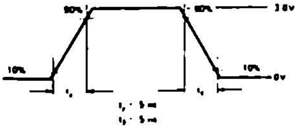
TEST LOAD CIRCUIT (THREE-STATE OUTPUTS)


ALL DIODES ARE 1N914 OR EQUIVALENT
VCC
S1
S2
TEST POINT
FROM OUTPUT UNDER TEST
CL
L H S1 OPEN
Z H S2 CLOSED
H L S1 CLOSED
Z L S2 OPEN
L Z S1 CLOSED
H Z S2 CLOSED

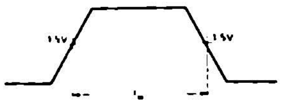
NOTE C_L includes fixture capacitance

VOLTAGE WAVEFORMS

INPUT WAVEFORM

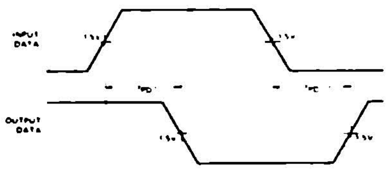

90%
90%
3.0V
10%
10%
0V
tr
tf
tr = 5ns
tf = 5ns

CLOCK PULSE WIDTH

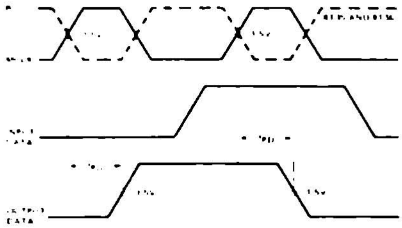


VOLTAGE WAVEFORMS (Cont'd)

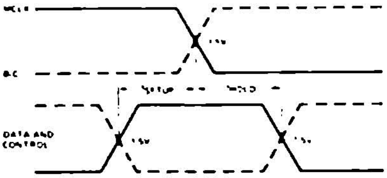
DATA DELAY TIMES
Input Data Reference


INPUT DATA
1.5V
1.5V
tPD
tPD
OUTPUT DATA
1.5V
1.5V

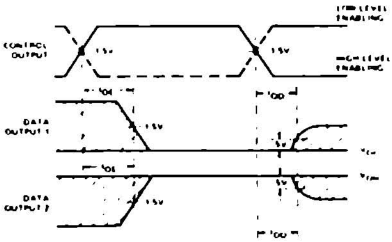
DATA DELAY TIMES
Clock Referenced


1.5V
1.5V
INPUT DATA
tPD
tPD
OUTPUT DATA
1.5V
1.5V

SETUP AND HOLD TIMES


MCLK
1.5V
S/C
SETUP
HOLD
DATA AND CONTROL
1.5V
1.5V

OUTPUT ENABLE AND DISABLE TIMES
(Three-State Outputs)


LOW LEVEL ENABLING
CONTROL OUTPUT
1.5V
1.5V
HIGH LEVEL ENABLING
tOE
tOD
DATA OUTPUT 1
1.5V
VOL
tOE
VOH
DATA OUTPUT 2
1.5V
tOD

Waveform #1 is for an output with internal conditions such that the output is Low when the three-state driver is enabled. Waveform #2 is for the opposite condition.

ADDRESS PROGRAMMING

The I/O Port is manufactured such that an address of all high levels (>2V) on the Microprocessor Bus inputs matches the Port's internal address. To program a bit so a low-level input (<0.8V) matches, the following procedure should be used.

1. Set all control inputs to their inactive state ($\overline{BIC}$ = $\overline{BOC}$ = $\overline{ME}$ = V_{CC}, SC = WC = MCLK = GND). Leave all Microprocessor Bus I/O pins open.
2. Raise V_{CC} to 7.75V ± .25V.
3. After V_{CC} has stabilized, apply a single programming pulse to the user data bus bit where a low-level match is desired. The voltage should be limited to 18V, the current should be limited to 75mA. Apply the pulse as shown in Figure 1.
4. Return V_{CC} to 0V. (Note 1)
5. Repeat this procedure for each bit where a low-level match is desired.
6. Verify that the proper address is programmed by setting the Port's status latch ($\overline{IV0\text{-}IV7}$ = desired address, $\overline{ME}$ = WC = L, SC = MCLK = H). If the proper address has been programmed, data presented at the µP bus will appear inverted on the user bus outputs. (Use normal V_{CC} and input voltage for verification.)

After the desired address has been programmed, a second procedure must be followed to isolate the address circuitry. The procedure is:

ADDRESS PROGRAMMING PULSE

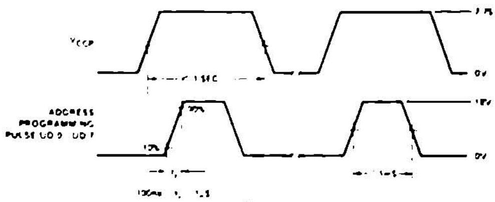


Figure 1

PROTECT PROGRAMMING PULSE

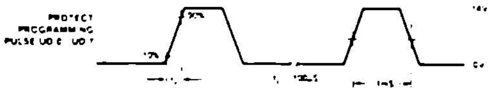


Figure 2

1. Set V_{CC} and all control inputs to 0V. (V_{CC} = $\overline{BIC}$ = $\overline{BOC}$ = $\overline{ME}$ = SC = WC = MCLK = 0V). Leave all Microprocessor Bus I/O pins open.
2. Apply a protect programming pulse to every user data bus pin, one at a time. The voltage should be limited to 14V, the current should be limited to 150mA. Apply the pulse as shown in Figure 2.
3. Verify that the address circuitry is isolated by applying 7V to each user data bus pin and measuring less than 1mA of input current. The conditions should be the same as in step 1 above. The rise time on the verification voltage must be slower than 100µs.

PROGRAMMING SPECIFICATIONS[1]

	PARAMETER	TEST CONDITIONS	LIMITS			UNITS
			Min	Typ	Max	
V_{CCP}	Programming supply voltage					
	Address		7.5		8.0	V
	Protect			0		V
I_{CCP}	Programming supply current	V_{CCP} = 8.0V			250	mA
	Max time V_{CCP} > 5.25V				1.0	s
	Programming voltage					
	Address		17.5		18.0	V
	Protect		13.5		14.0	V
	Programming current					
	Address				75	mA
	Protect				150	mA
	Programming pulse rise time					
	Address		1		1	µs
	Protect		100			µs
	Programming pulse width		5		1	ms

NOTE

1. If all programming can be done in less than 1 second, V_{CC} may remain at 7.75V for the entire programming cycle.

APPLICATIONS

Figure 3 shows some of the various ways to use the I/O Port in a system. By controlling the $\overline{\text{BIC}}$ and $\overline{\text{BOC}}$ lines, the device may be used for the input and output of data, control, and status signals. I/O Port 1 functions bidirectionally for data transfer and I/O Port 2 provides a similar function for discrete status and control lines. I/O Ports 3 and 4 serve as dedicated output and input ports, respectively.

APPLICATIONS

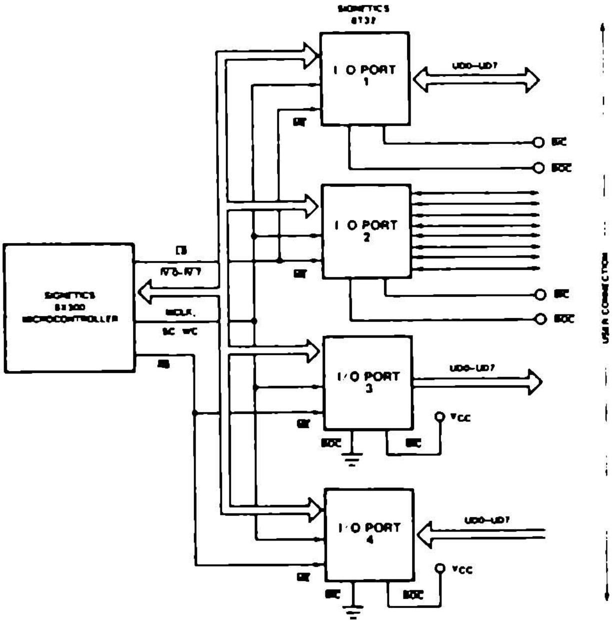

SIGNETICS
8T32
I/O PORT 1
UD0–UD7
ME
BIC
BOC
I/O PORT 2
ME
BIC
BOC
SIGNETICS
8X300
MICROCONTROLLER
IV0–IV7
MCLK,
SC WC
I/O PORT 3
UD0–UD7
VCC
ME
BOC
BIC
I/O PORT 4
UD0–UD7
VCC
ME
BIC
BOC
USER CONNECTION

Figure 3

VALVO

DESCRIPTION

The Bus Expander is specifically designed to increase the I/O capability of 8X300 systems previously limited by fanout considerations. The bus expander serves as a buffer between the 8X300 and blocks of I/O devices. Each bus expander can buffer a block of 16 I/O ports while only adding a single load to the 8X300.

FEATURES

- **15ns propagation delay**
- **Bidirectional**
- **Three-state outputs on both ports**
- **Pre-programmed address range**

APPLICATIONS

The 8T39 Bus Expander is designed to be used with the 8X300 microprocessor to allow increased I/O capability in those systems previously limited by fanout considerations. Figure 1 shows a typical arrangement of the bus expander in an 8X300 system. Each expander services I/O ports whose address is within the range of the expander. Other I/O ports or working storage may be directly connected to the bus as shown.

The bus expander is not limited to use with the 8X300, but may be applied in any system which uses a combined address/data bus.

8T39 ADDRESSING

During normal operation of the 8X300 when an I/O port address is being sent on the IV Bus (SC is high), the I/O port will examine all eight bits of the IV Bus for an address compare. Since the 8T39 is used to buffer blocks of I/O ports, only the four most significant bits are examined by the 8T39 for an address compare•

Note that redundant addresses are not programmed into separate devices. Rather, a discrete device (such as the 8T39-03) may be wired for any address requiring two 1 bits and two 0 bits in the address. The various address ranges for this same device are obtained by permuting the high order (DI0 and DO0 are MSB) data lines accordingly. Both input and output lines must be redefined in order to maintain data and address integrity on the extended bus. Table 1 summarizes the 8T39 addressing.

ORDERING INFORMATION

The Bus Expander is ordered by specifying the following part number:

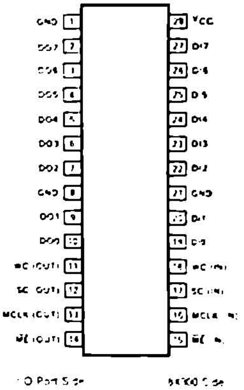
N8T39-XXP
P = { I - Ceramic Package
XL - Epoxy Package
Address Range As Determined From Table 1

Address functions are specified with the convention that bit 0 is the MSB and bit 7 is the LSB. The DI bus address decoding is active low.

FUNCTIONAL DESCRIPTION

The Bus Expander contains eight sets of non-inverting bidirectional tri-state drivers for the bus data bits, four non-inverting unidirectional drivers for I/O port control and necessary control logic. The control logic is required to maintain the proper directional transfer of bus data as dictated by the states of the I/O port control signals and the currently enabled I/O port. Each bus expander is programmed during manufacturing to respond to a specific block of I/O port addresses. Only I/O ports with addresses in the range of a given bus expander may be connected to that expander. A bus expander may be used on either left bank or right bank. Multiple expanders on the same bank must have different address ranges, however, expanders with the same address range can be connected if they are on different banks. Systems may be configured with I/O ports connected directly to the 8X300, as well as I/O ports connected through a bus expander, however, no unbuffered I/O port may have an address within the span of a bus expander on the same bank.

PIN CONFIGURATION



PIN DESIGNATION

PIN NO.	SYMBOL	NAME & FUNCTION	TYPE
2-7,9,10	DO0-DO7	I/O port data bus	Active low, three-state
11	WC(OUT)	Write control output	Active high
12	SC(OUT)	Select control output	Active high
13	MCLK(OUT)	Master clock output	Acitve high
14	$\overline{\text{ME}}$(OUT)	Master enable output	Active low
15	$\overline{\text{ME}}$(IN)	Master enable input	Active low
16	MCLK	Master clock input	Active high
17	SC(IN)	Select control input	Active high
18	WC(IN)	Write control input	Active high
19,20,22-27	DI0-DI7	Microcontroller data bus	Active low, three-state
1,8,21	GND	Ground	Active low, three-state
28	VCC	+5 volt supply	

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	-0.5 to +7	Vdc
V_{IN}	Input voltage	-0.5 to +5.5	Vdc
V_O	Off-state output voltage	-0.5 to +5.5	Vdc
T_A	Operating temperature range	0 to +70	°C
T_{stg}	Storage temperature range	-65 to +150	°C

Addition of bus expanders may impact system cycle time due to the added delay in the data path. For the purposes of calculating allowable cycle time as described in the 8X309 data sheet, the bus expander delays may be considered additive to the I/O port delays so that a buffered I/O port simply appears as a slower I/O port.

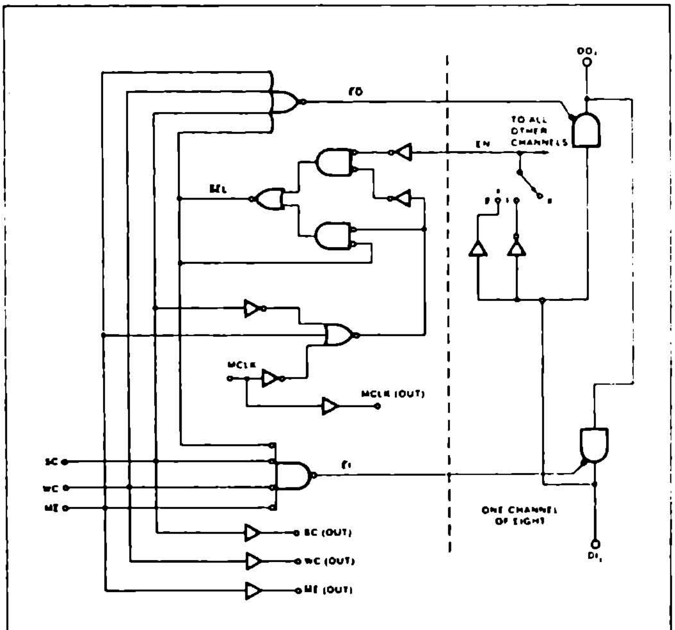
TRUTH TABLE

$\overline{ME}$	SC	WC	MCLK	SELECT LATCH	DATA TRANSFER DIRECTION	ADDRESS* COMPARISON
L	L	L	X	Set	DI Bus - DO Bus	No
L	L	L	X	Not set	DI Bus - DO Bus	No
L	L	H	X	X	DI Bus - DO Bus	No
L	H	X	L	X	DI Bus - DO Bus	No
L	H	X	H	X	DI Bus - DO Bus	Yes
H	X	X	X	X	DI Bus - DO Bus	No

NOTES

*When an address comparison is made, the select latch is set if the data on the DI Bus is within the manufactured address range of the I/O Bus Expander. Otherwise, the select latch is cleared.

BLOCK DIAGRAM



NOTES

*Selection made during manufacture

X Don't care

PART TYPE	ADDRESS PATTERN MSB(0) LSB(7)	ADDRESS BLOCKS (Decimal)
8T39-00	0000XXXX	0-15
8T39-01	0001XXXX	16-31, 32-47, 64-79, 128-143
8T39-03	0011XXXX	48-63, 80-95, 144-159, 96-111, 160-175, 192-207
8T39-07	0111XXXX	112-127, 176-191, 208-223, 224-239
8T39-17	1111XXXX	240-255

Table 1 8T39 ADDRESSING SUMMARY

TYPICAL APPLICATION USING 2 BUS EXPANDERS TO GIVE 33 I/O PORTS PLUS WORKING STORAGE

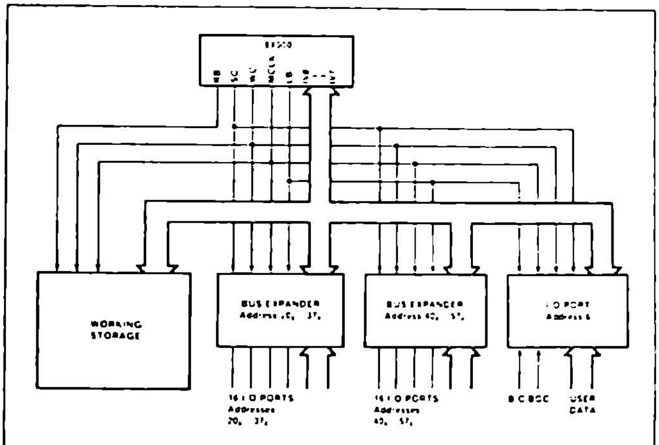
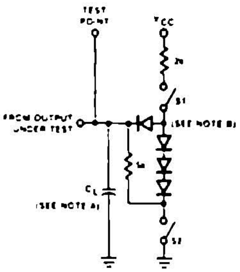

WC
SC
MCLK
WORKING STORAGE
BUS EXPANDER
Address 20₈–37₈
BUS EXPANDER
Address 40₈–57₈
I/O PORT
Address 6
16 I/O PORTS
Addresses
20₈–37₈
16 I/O PORTS
Addresses
40₈–57₈
USER
DATA

Figure 1

TEST LOAD CIRCUIT


TEST POINT
VCC
S1
FROM OUTPUT UNDER TEST
(SEE NOTE B)
CL
(SEE NOTE A)
S2

NOTES

A C_L includes probe and jig capacitance

B All diodes are 1N916 or 1N3064

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%, 0°C \leq T_A \leq 70°C$

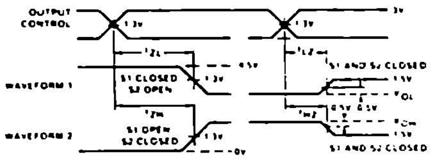
	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				.8	
V_{IH}	High		2.0			
V_{IC}	Clamp				-1	
	Output voltage	V_{CC} = 4.75V				V
V_{OL}	Low	I_{OL} = 16mA			.55	
V_{OH}	High	I_{OH} = -3.2mA	2.4			
	Input current	V_{CC} = 5.25V				uA
I_{IL}	Low	V_{IL} = .5V			-250	
I_{IH}	High	V_{IH} = 5.25V		< 10	100	
I_{OS}	Short circuit output current	V_{CC} = 4.75V	-40			mA
I_{CC}	Supply current	V_{CC} = 5.25V			200	mA

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%, 0°C \leq T_A \leq 70°C, C_L = 300pF$

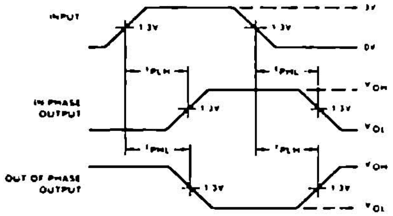
PARAMETER		TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
	Path delay							ns
tpd	Data	DOX DIX	DIX DOX				15	
tpd	Control	$\overline{ME}$ (out) MCLK (out) SC (out WC (out)	$\overline{ME}$ (in) MCLK (in) SC (in) WC (in)				15	

VOLTAGE WAVEFORMS

CONTROL PATH DELAY
(THREE-STATE OUTPUTS


OUTPUT CONTROL
1.3V
3V
1.3V
tZL
4.5V
S1 CLOSED
S2 OPEN
1.3V
WAVEFORM 1
tZH
S1 OPEN
S2 CLOSED
1.3V
WAVEFORM 2
0V
tLZ
S1 AND S2 CLOSED
1.5V
VOL
0.5V 0.5V
tHZ
VOH
1.5V
S1 AND S2 CLOSED

DATA PATH DELAY TIMES


INPUT
1.3V
1.3V
3V
0V
tPLH
tPHL
IN PHASE OUTPUT
VOH
1.3V
1.3V
VOL
tPHL
tPLH
OUT OF PHASE OUTPUT
VOH
1.3V
1.3V
VOL

DESCRIPTION

The Bus Expander is specifically designed to increase the I/O capability of 8X300 systems previously limited by fanout considerations. The bus expander serves as a buffer between the 8X300 and blocks of I/O devices. Each bus expander can buffer a block of 16 I/O ports while only adding a single load to the 8X300.

FEATURES

- **15ns max propagation delay**
- **Bidirectional**
- **Three-state outputs on both ports**

FUNCTIONAL DESCRIPTION

The Bus Expander contains eight sets of non-inverting bidirectional tri-state drivers for the bus data bits, four non-inverting undirectional drivers for I/O port control, and necessary control logic. The control logic is required to maintain the proper directional transfer of bus data as dictated by the states of the I/O port control signals. A bus expander may be used on either left bank or right bank. Systems may be configured with I/O ports connected directly to the 8X300, as well as I/O ports connected through a bus expander.

Addition of bus expanders may impact system cycle time due to the added delay in the data path. For the purposes of calculating allowable cycle time as described in the 8X300 data sheet, the bus expander delays may be considered additive to the I/O port delays so that a buffered I/O port simply appears as a slower I/O port.

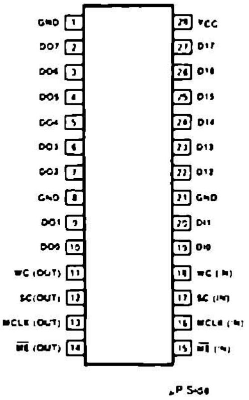
APPLICATIONS

The 8T39 Bus Expander is designed to be used with the 8X300 microprocessor to allow increased I/O capability in those systems previously limited by fanout considerations. Figure 1 shows a typical arrangement of the bus expander in an 8X300 system. Other I/O ports or working storage may be directly connected to the bus as shown.

The bus expander is not limited to use with the 8X300, but may be applied in any system which uses a combined address/data bus.

PIN CONFIGURATION

I,XL PACKAGE



TRUTH TABLE

$\overline{ME}$	SC	WC	DATA TRANSFER DIRECTION	ADDRESS COMPARISON
L	L	L	DI Bus — DO Bus	No
L	L	H	DI Bus — DO Bus	No
L	H	X	DI Bus — DO Bus	No
H	X	X	DI Bus — DO Bus	No

PIN DESIGNATION

PIN NO.	SYMBOL	NAME & FUNCTION	TYPE
2-7,9,10	DO0-DO7	I/O port data bus	Active low, three-state
11	WC(OUT)	Write command output	Active high
12	SC(OUT)	Select command output	Active high
13	MCLK(OUT)	Master clock input	Active high
14	$\overline{ME}$(OUT)	Master enable output	Active low
15	$\overline{ME}$(IN)	Master enable input	Active low
16	MCLK(IN)	Master clock input	Active high
17	SC(IN)	Select command output	Active high
18	WC(IN)	Write command output	Active high
19,20,22-27	DI0-DI7	Microprocessor data bus	Active low, three-state
1,8,21	GND	Ground	
28	Vcc	+5 volt supply	

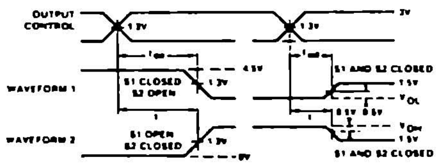
DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $0°C \le T_A \le 70°C$

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				8	
V_{IH}	High		2.0			
V_{IC}	Clamp	-5mA at V_{CC} min			-1	
	Output voltage	V_{CC} = 4.75V				V
V_{OL}	Low	I_{OL} · 50mA			.55	
V_{OH}	High	I_{OH} = -3 2mA	2.4			
	Input current	V_{CC} = 5 25V				μA
I_{IL}	Low*	V_{IL} = 5V			-250	
I_{IH}	High*	V_{IH} = 5 25V		<10	100	
I_{OS}	Short circuit output current	V_{CC} = 4.75V	-40			mA
I_{CC}	Supply current	V_{CC} = 5.25V			200	mA

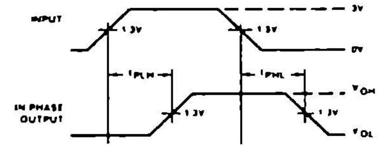
*Includes 3-State leakage

VOLTAGE WAVEFORMS

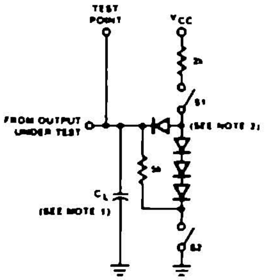
PROPAGATION DELAY TO THREE-STATE OUTPUTS


OUTPUT CONTROL
1.3V
1.3V
3V
WAVEFORM 1
WAVEFORM 2
S1 CLOSED S2 OPEN
S1 OPEN S2 CLOSED
4.5V
1.3V
0V
S1 AND S2 CLOSED
1.5V
V_{OL}
0.5V 0.5V
V_{OH}
1.5V
S1 AND S2 CLOSED

PROPAGATION DELAY TIMES


INPUT
1.3V
1.3V
3V
0V
t_{PLH}
t_{PHL}
IN PHASE OUTPUT
1.3V
1.3V
V_{OH}
V_{OL}

TEST LOAD CIRCUIT


TEST POINT
V_{CC}
2k
S1
FROM OUTPUT UNDER TEST
(SEE NOTE 2)
5k
C_L
(SEE NOTE 1)
S2

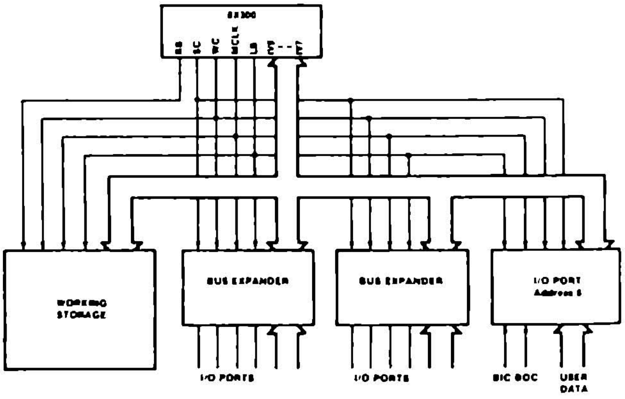
All resistors values are typical and in ohms

NOTES

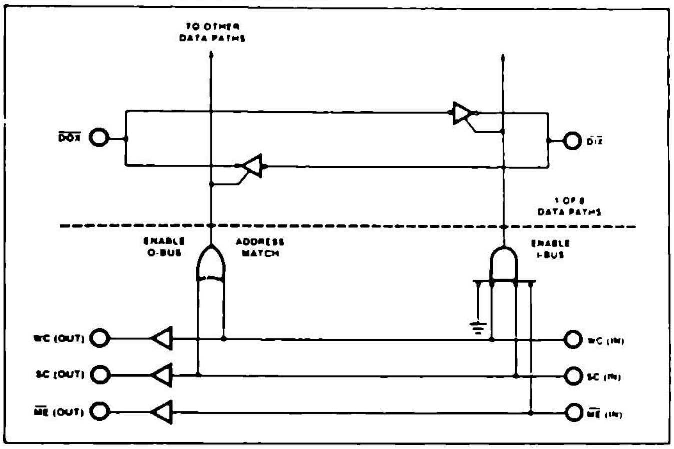
1. C_L includes probe and jig capacitance
2. All diodes are 1N916 or 1N3064

TYPICAL APPLICATION

USING 2 BUS EXPANDERS
TO CREATE 33 I/O PORTS PLUS WORKING STORAGE


8X300
RB
SC
WC
MCLK
LB
IV0
IV7
WORKING STORAGE
BUS EXPANDER
BUS EXPANDER
I/O PORT Address 5
I/O PORTS
I/O PORTS
SIC SOC
USER DATA

FUNCTIONAL BLOCK DIAGRAM


TO OTHER DATA PATHS
$\overline{DOX}$
$\overline{DIX}$
1 OF 8 DATA PATHS
ENABLE O-BUS
ADDRESS MATCH
ENABLE I-BUS
WC (OUT)
SC (OUT)
$\overline{ME}$ (OUT)
WC (IN)
SC (IN)
$\overline{ME}$ (IN)

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
V_O	Off-state output voltage	+5.5	Vdc
T_A	Operating temperature range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $0°C \le T_A \le 70°C$, $C_L = 300pF$

PARAMETER		TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
t_{pd}	Path delay Data	DOX DIX	DIX DOX				15	ns
t_{pd}	Control	$\overline{ME}$(OUT) MCLK(OUT) SC(OUT) WC(OUT)	$\overline{ME}$(IN) MCLK(IN) SC(IN) WC(IN)				15	ns
t_{oe}	Data Output Enable	DIX DOX	$\overline{ME}$(IN) SC(IN) WC(IN)		28		56	ns
t_{od}	Data Output Disable	DIX DOX	$\overline{ME}$(IN) SC(IN) WC(IN)		15			

4096-BIT BIPOLAR PROM (1024X4) 82HS137 (T.S.)

82HS137I-F,N,R

DESCRIPTION

The 82HS137 is field programmable, which means that custom patterns are immediately available by following the fusing procedure given in this data manual. The 82HS137 is supplied with all outputs at logical low. Outputs are programmed to a logic high level at any specified address by fusing a Ni-Cr link matrix.

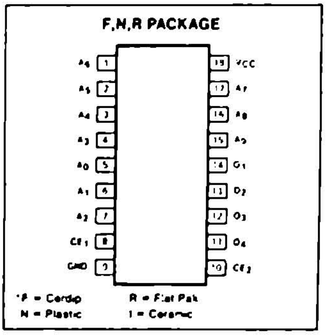
These devices include on-chip decoding and 2 chip enable inputs for ease of memory expansion. They feature tri-state outputs for optimization of word expansion in bused organizations.

The 82HS137 device is available in the commercial and military temperature ranges. For the commercial temperature range (0°C to +75°C) specify N82HS137, F or N, and for the military temperature range (-55°C to +125°C) specify S82HS137, I, F, or R.

FEATURES

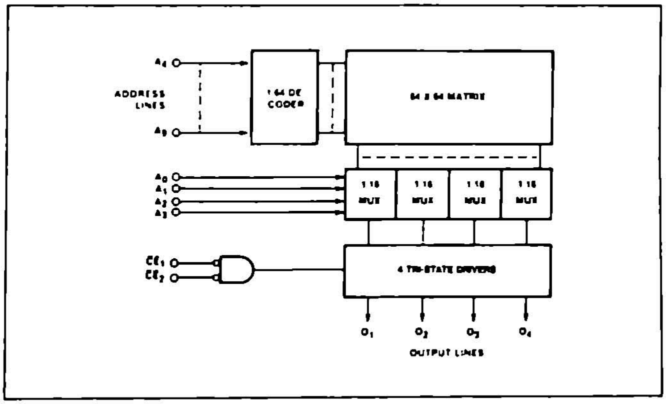
- Address access time:
  - N82HS137: 45ns max
  - S82HS137: 70ns max
- Power dissipation: 13mW/bit typ
- Input loading:
  - N82HS137: −100μA max
  - S82HS137: −150μA max
- On-chip address decoding
- No separate fusing pins
- Unprogrammed outputs are low level
- Fully TTL compatible

PIN CONFIGURATION


F,N,R PACKAGE
A6 1
A5 2
A4 3
A3 4
A0 5
A1 6
A2 7
CE1 8
GND 9
18 VCC
17 A7
16 A8
15 A9
14 O1
13 O2
12 O3
11 O4
10 CE2

*F = Cerdip R = Flat Pak
N = Plastic I = Ceramic

BLOCK DIAGRAM


A4
ADDRESS LINES
A9
1:64 DECODER
64 X 64 MATRIX
A0
A1
A2
A3
1:16 MUX
1:16 MUX
1:16 MUX
1:16 MUX
CE1
CE2
4 TRI-STATE DRIVERS
O1
O2
O3
O4
OUTPUT LINES

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
	Output voltage		Vdc
V_O	Off-state	+5.5	
	Temperature range		°C
T_A	Operating		
	N82HS137	0 to +75	
	S82HS137	−55 to +125	
T_{STG}	Storage	−65 to +150	

DC ELECTRICAL CHARACTERISTICS

N82HS137: 0°C $\leq T_A \leq$ +75°C, 4.75V $\leq V_{CC} \leq$ 5 25V
S82HS137: −55°C $\leq T_A \leq$ +125°C, 4.5V $\leq V_{CC} \leq$ 5.5V

PARAMETER		TEST CONDITIONS	N82HS137			S82HS137			UNIT
			Min	Typ[2]	Max	Min	Typ	Max	
	Input voltage								V
V_{IL}	Low				.85			.80	
V_{IH}	High		2.0			2.0			
V_{IC}	Clamp	I_{IN} = −18mA		−0.8	−1.2			−1.2	
	Output voltage	$\overline{CE}_{1,2}$ = Low							V
V_{OL}	Low	I_{OUT} = 16mA			0.45			0.5	
V_{OH}	High	I_{OUT} = −2mA	2.4			2.4			
	Input current								μA
I_{IL}	Low	V_{IN} = 0.45V			−100			−150	
I_{IH}	High	V_{IN} = 5.5V			40			50	
	Output current								
$I_{O(OFF)}$	Hi-Z state	$\overline{CE}_{1,2}$ = High, V_{OUT} = 0.5V			40			60	μA
		$\overline{CE}_{1,2}$ = High, V_{OUT} = 5.5V			−40			−60	μA
I_{OS}	Short circuit	$\overline{CE}_{1,2}$ = Low, V_{OUT} = 0V,			40			60	
		Stored High	−20		−70	−15		−85	mA
I_{CC}	V_{CC} supply current			85	140			150	mA
	Capacitance	$\overline{CE}_{1,2}$ = High, V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V		8			8		

AC ELECTRICAL CHARACTERISTICS

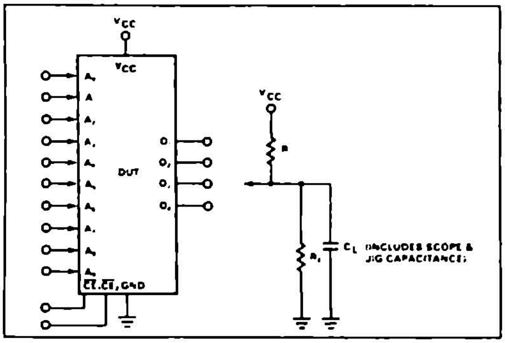
R_1 = 270Ω, R_2 = 600Ω, C_L = 30pF 1
N82HS137: 0°C $\leq T_A \leq$ +75°C, 4.75V $\leq V_{CC} \leq$ 5.25V
S82S137; S82S137A: −55°C $\leq T_A \leq$ +125°C, 4.5V $\leq V_{CC} \leq$ 5.5V

PARAMETER		TO	FROM	N82HS137			S82HS137			UNIT
				Min	Typ[2]	Max	Min	Typ	Max	
	Access time									ns
T_{AA}		Output	Address		35	45			70	
T_{CE}		Output	Chip enable		15	30			40	
	Disable time									ns
T_{CD}		Output	Chip disable		15	30			40	

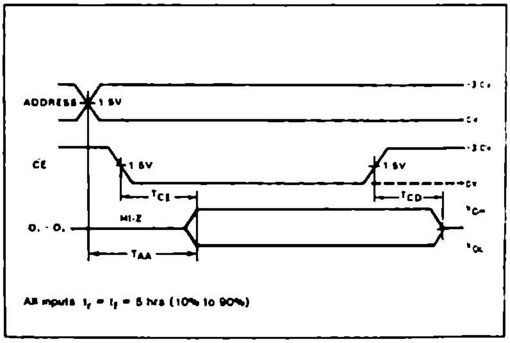
Note

1. Positive current is defined as into the terminal referenced
2. Typicals are at Room Temperature 5V

TEST LOAD CIRCUIT


V_{CC}
V_{CC}
DUT
A_0 … A_9
$\overline{CE}$, $\overline{CE}$, GND
O_1
O_2
O_3
O_4
V_{CC}
R_1
R_2
C_L (INCLUDES SCOPE & JIG CAPACITANCE)

VOLTAGE WAVEFORM


ADDRESS
1.5V
+3.0V
0V
$\overline{CE}$
1.5V
1.5V
+3.0V
0V
T_{CE}
T_{CD}
HI-Z
O_1 - O_4
V_{OH}
V_{OL}
T_{AA}

All inputs $t_r = t_f$ = 5 ns (10% to 90%)

4096-BIT BIPOLAR PROM (512X8) 82HS147 (T.S.)

82HS147-F,N

DESCRIPTION

The 82HS147 is field-programmable, which means that custom patterns are immediately available by following the fusing procedure given in this data manual. The standard devices are supplied with all outputs at logical low. Outputs are programmed to a logic high level at any specified address by fusing a Ni-Cr link matrix.

The 82HS147 includes on-chip decoding and one chip enable input for ease of memory expansion, and features tri-state outputs for optimization of word expansion in bused organizations.

The 82HS147 device is available in the commercial temperature range (0°C to +75°C), and is specified as N82HS147, F, N.

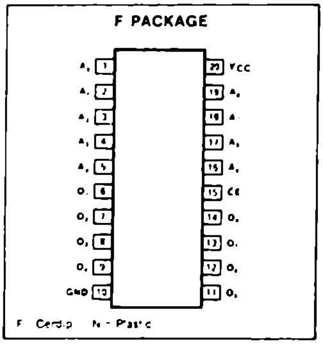
FEATURES

- **Address access time: 45ns max**
- **Power dissipation: 853mW max**
- **Input loading: -100μA max**
- **One chip enable input**
- **On chip address decoding**
- **No separate fusing pins**
- **Fully TTL compatible**

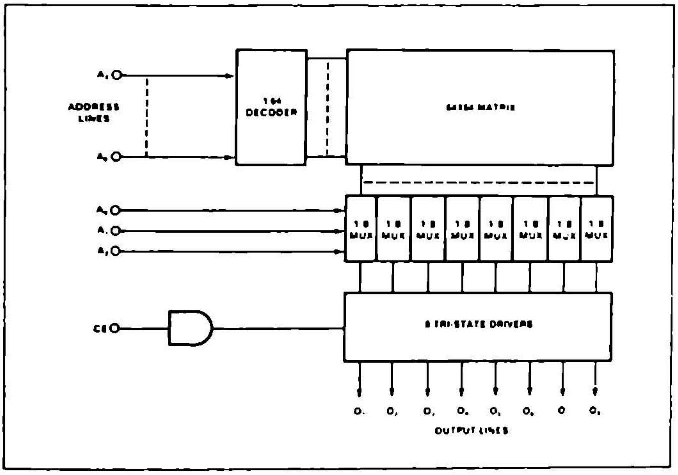
APPLICATIONS

- **Prototyping/volume production**
- **Sequential controllers**
- **Microprogramming**
- **Hardwired algorithms**
- **Control store**
- **Random logic**
- **Code conversion**

PIN CONFIGURATION



BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
	Output voltage		Vdc
V_O	Off-state	+5.5	°C
	Temperature range		
T_A	Operating	0 to +75	
T_{STG}	Storage	-65 to +150	

OBJECTIVE SPECIFICATION 82HS147-F,N

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C, 4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1]	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage[2]					V
V_{IL}	Low				.85	
V_{IH}	High		2.0			
V_{IC}	Clamp	I_{IN} = -18mA			-1.2	
	Output voltage					V
V_{OL}	Low	I_{OUT} = 9.6mA			0.45	
V_{OH}	High	$\overline{CE}$ = Low, I_{OUT} = -2mA, High stored	2.4			
	Input current					µA
I_{IL}	Low	V_{IN} = 0.45V			-100	
I_{IH}	High	V_{IN} = 5.5V			40	
	Output current					
I_{OLK}	Leakage	$\overline{CE}$ = High, V_{OUT} = 5.5V			40	µA
$I_{O(OFF)}$	Hi-Z state	$\overline{CE}$ = High, V_{OUT} = 0.5V			-40	µA
		$\overline{CE}$ = High, V_{OUT} = 5.5V			40	
I_{OS}	Short circuit[3]	V_{OUT} = 0V	-20		-70	mA
I_{CC}	V_{CC} supply current				155	mA
	Capacitance	V_{CC} = 5.0V, $\overline{CE}$ = High				pF
C_{IN}	Input	V_{IN} = 2.0V		5		
C_{OUT}	Output	V_{OUT} = 2.0V		8		

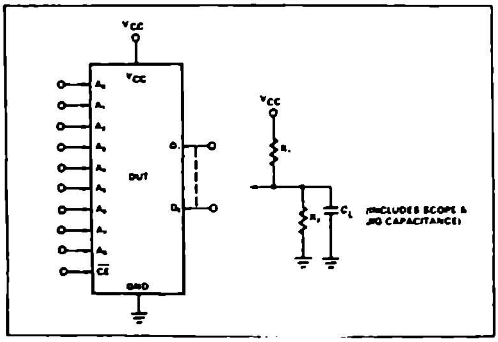
AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega, R_2 = 1k\Omega, C_L = 30pF, 0°C \leq T_A \leq +75°C, 4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TO	FROM	LIMITS			UNIT
				Min	Typ	Max	
	Access time						ns
T_{AA}[4]		Output	Address			45	
T_{CE}		Output	Chip enable			30	
T_{CD}	Disable time	Output	Chip disable			30	ns

NOTES

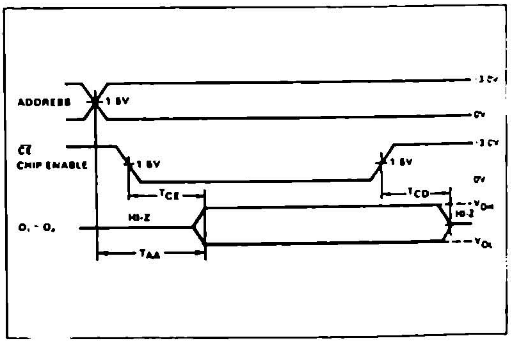
1. Positive current is defined as into the terminal referenced
2. All voltages with respect to network ground
3. Duration of short circuit should not exceed 1 second
4. Tested at an address cycle time of 1µ sec

TEST LOAD CIRCUIT



V_{CC}
A_0
A_1
A_2
A_3
A_4
A_5
A_6
A_7
A_8
$\overline{CE}$
V_{CC}
OUT
GND
O_1
O_8
V_{CC}
R_1
R_2
C_L
(INCLUDES SCOPE & JIG CAPACITANCE)

VOLTAGE WAVEFORM



ADDRESS
1.5V
3.0V
0V
$\overline{CE}$
CHIP ENABLE
1.5V
3.0V
1.5V
0V
T_{CE}
T_{CD}
Hi-Z
$O_1 - O_8$
T_{AA}
Hi-Z
V_{OH}
V_{OL}

8192-BIT BIPOLAR PROM (1024X8) 82HS181 (T.S.)

PRELIMINARY DATA SHEET

82HS181-F,N,R,G,I

DESCRIPTION

The 82HS180 and 82HS181 are field programmable, which means that custom patterns are immediately available by following the fusing procedure given in this data manual. The 82HS180 and 82HS181 are supplied with all outputs at logical low. Outputs are programmed to a logic high level at any specified address by fusing a Ni-Cr link matrix.

These devices include on-chip decoding and 4 chip enable inputs for ease of memory expansion. They feature either open collector or tri-state outputs for optimization of work expansion in bused organizations.

The 82HS180 and 82HS181 are available in both the commercial and military temperature ranges. For the commercial temperature range (0° to +75°C) specify N82HS181 I, F or N, and for the military temperature range (-55°C to +125°C) specify S82HS181, R, F, G, I.

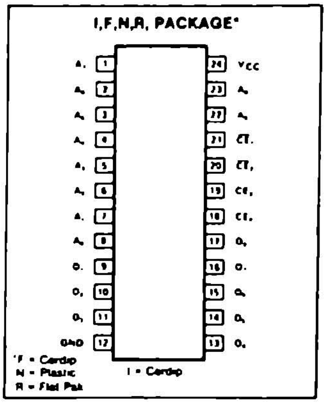
FEATURES

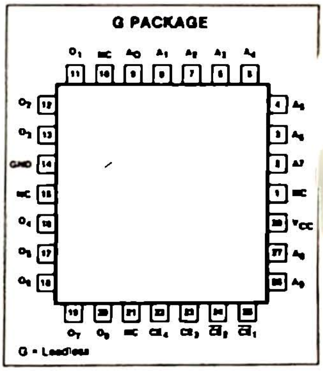
- **Address access time:**
  - N82HS180/181: 50ns max
  - S82HS180/181: 70ns max
- **Power dissipation: 85μW/bit typ**
- **Input loading:**
  - N82HS180/181: -100μA max
  - S82HS180/181: -150μA max
- **On-chip address decoding**
- **Output options:**
  - 82HS 181: Tri-State
- **No separate fusing pins**
- **Unprogrammed outputs are low level**
- **Fully TTL compatible**

APPLICATIONS

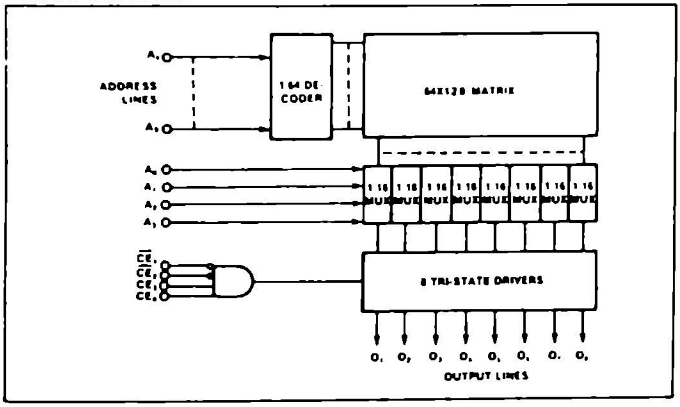
- **Prototyping/volume production**
- **Sequential controllers**
- **Microprogramming**
- **Hardwired algorithms**
- **Control store**
- **Random logic**
- **Code conversion**

PIN CONFIGURATION


I,F,N,R, PACKAGE*
1
2
3
4
5
6
7
8
9
10
11
GND 12
24 VCC
23
22
21
20
19
18
17
16
15
14
13
*F = Cerdip
N = Plastic
R = Flat Pak
I = Cerdip


G PACKAGE
GND 14
VCC
G = Leadless

BLOCK DIAGRAM


ADDRESS LINES
1:64 DECODER
64X128 MATRIX
1:16 MUX
8 TRI-STATE DRIVERS
OUTPUT LINES

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
Vcc	Supply voltage	+7	Vdc
VIN	Input voltage	+5.5	Vdc
	Output voltage		Vdc
Vo	Off-state	+5.5	
	Temperature range		°C
TA	Operating		
	N82HS181	0 to +75	
	S82HS181	-55 to +125	
TSTG	Storage	-65 to +150	

DC ELECTRICAL CHARACTERISTICS

N82HS181: 0°C ≤ T_A ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V
S82HS181: -55°C ≤ T_A ≤ +125°C, 4.5V ≤ V_{CC} ≤ 5.5V

PARAMETER		TEST CONDITIONS[1]	N82HS181:			S82HS181:			UNIT
			Min	Typ[3]	Max	Min	Typ	Max	
	Input voltage								V
V_{IL}	Low				85			80	
V_{IH}	High		2.0			2.0			
V_{IC}	Clamp	I_{IN} = -18mA		-0.8	-1.2			-1.2	
	Output voltage	$\overline{CE}_{1,2}$ = Low, $CE_{3,4}$ = High							V
V_{OL}	Low	I_{OUT} = 9.6mA			0.45			0.5	
V_{OH}	High	I_{OUT} = -2mA	2.4			2.4			
	Input current								µA
I_{IL}	Low	V_{IN} = 0.45V			-100			-150	
I_{IH}	High	V_{IN} = 5.5V			40			50	
	Output current								
I_{OLK}									
$I_{O(OFF)}$	Hi-Z state	$\overline{CE}_{1,2}$ = High, $CE_{3,4}$ = Low, V_{OUT} = 5.5V			40			-60	µA
		$\overline{CE}_{1,2}$ = High, $CE_{3,4}$ = Low, V_{OUT} = 0.5V			-40			60	µA
I_{OS}	Short circuit	$\overline{CE}_{1,2}$ = Low, $CE_{3,4}$ = High, V_{OUT} = 0V High Stored	-20		-70	-15		-85	mA
I_{CC}	V_{CC} supply current	$\overline{CE}_{1,2}$ = Low, $CE_{3,4}$ = High		125	175			185	mA
	Capacitance	$\overline{CE}_{1,2}$ = High, V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V		8			8		

AC ELECTRICAL CHARACTERISTICS

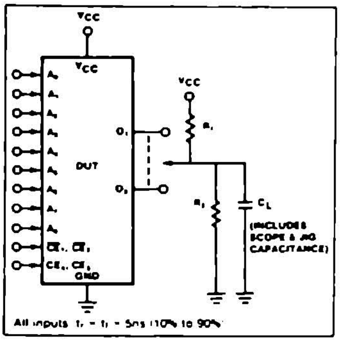
R_1 = 470Ω, R_2 = 1kΩ, C_L = 30pF
N82HS181: 0°C ≤ T_A ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V
S82HS181: -55°C ≤ T_A ≤ +125°C, 4.5V ≤ V_{CC} ≤ 5.5V

PARAMETER		TO	FROM	N82HS181:			S82HS181:			UNIT
				Min	Typ[3]	Max	Min	Typ	Max	
	Access time									ns
T_{AA}[2]		Output	Address		40	50			70	
T_{CE}		Output	Chip enable		20	35			45	
	Disable time									ns
T_{CD}		Output	Chip disable		20	35			45	

NOTES

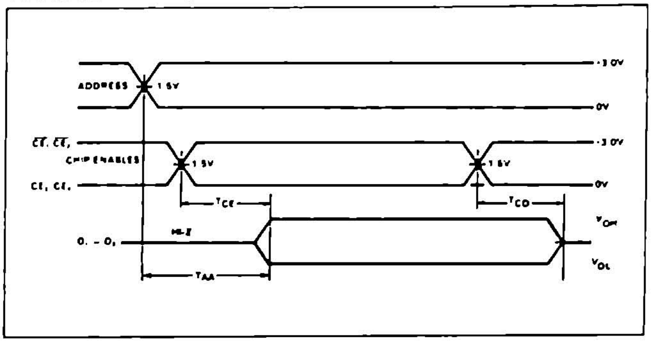
1. Positive current is defined as into the terminal referenced
2. Tested at an address cycle time of 1µsec.
3. Typical values are V_{CC} = 5V, T_A = +25

TEST LOAD CIRCUIT


VCC
VCC
A0
A1
A2
A3
A4
A5
A6
A7
A8
$\overline{CE}_1$, $\overline{CE}_2$
CE3, CE4
GND
DUT
O1
O8
VCC
R1
R2
CL
(INCLUDES SCOPE & JIG CAPACITANCE)

All inputs: $t_r = t_f = 5ns$ (10% to 90%)

VOLTAGE WAVEFORM


ADDRESS
1.5V
+3.0V
0V
$\overline{CE}_1$, $\overline{CE}_2$
CHIP ENABLES
CE3, CE4
1.5V
+3.0V
0V
1.5V
T_{CE}
T_{CD}
HI-Z
O1 – O8
V_{OH}
V_{OL}
T_{AA}

8192-BIT BIPOLAR PROM (2048X4) 82HS185 (T.S.)

Preliminary

82HS185-I,F,N,R

DESCRIPTION

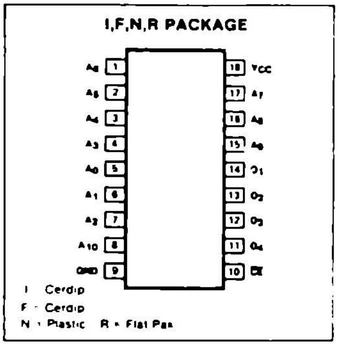
The 82HS185 is field programmable, which means that custom patterns are immediately available by following the fusing procedure given in this data manual. The standard 82HS185 is supplied with all outputs at logical low. Outputs are programmed to a logic high level at any specified address by fusing a Ni-Cr link matrix.

These devices include on-chip decoding and 1 chip enable input for memory expansion. It features tri-state outputs for optimization of word expansion in bused organizations.

The 82HS185 device is available in the commercial and military temperature ranges. For the commercial temperature range (0°C to +75°C) specify N82HS185, I, F N, and for the military temperature range (-55°C to +125°C) specify S82HS185, I, F, or R.

FEATURES

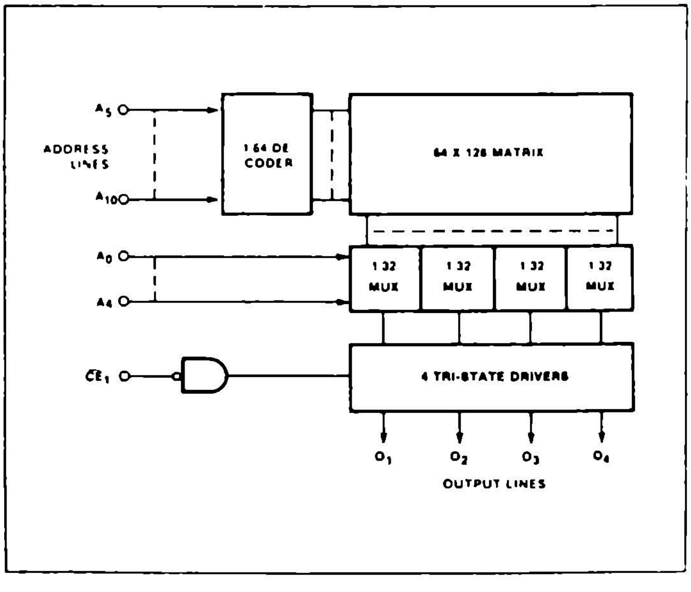
- **Low power dissipation: 50μW/bit typ**
- **Address access time:**
  - **N82HS185:** 50ns max
  - **S82HS185:** 70ns max
- **Input loading:**
  - **N82HS185: -100μA max**
  - **S82HS185: -150μA max**
- **On-chip address decoding**
- **No separate fusing pins**
- **Unprogrammed outputs are low level**
- **Fully TTL compatible**


I,F,N,R PACKAGE
A6 1
A5 2
A4 3
A3 4
A0 5
A1 6
A2 7
A10 8
GND 9
18 Vcc
17 A7
16 A8
15 A9
14 O1
13 O2
12 O3
11 O4
10 CE
I Cerdip
F - Cerdip
N - Plastic R = Flat Pak

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
	Output voltage		Vdc
V_O	Off-state	+5.5	
	Temperature range		°C
T_A	Operating		
	N82HS185	0 to +75	
	S82HS185	-55 to +125	
T_{STG}	Storage	-65 to +150	

BLOCK DIAGRAM


A5
ADDRESS LINES
A10
1:64 DECODER
64 X 128 MATRIX
A0
A4
1:32 MUX
1:32 MUX
1:32 MUX
1:32 MUX
CE1
4 TRI-STATE DRIVERS
O1
O2
O3
O4
OUTPUT LINES

8192-BIT BIPOLAR PROM (2048X4)

82HS185 (T.S.)

82HS185-I,F,N,R

Preliminary

DC ELECTRICAL CHARACTERISTICS

N82HS185: 0°C ≤ T_A ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V
S82HS185: -55°C ≤ T_A ≤ +125°C, 4.5V ≤ V_{CC} ≤ 5.5V

PARAMETER		TEST CONDITIONS[1] [2]	N82HS185			S82HS185			UNIT
			Min	Typ[5]	Max	Min	Typ	Max	
	Input voltage[1]								V
V_{IL}	Low				85			80	
V_{IH}	High		2.0			2.0			
V_{IC}	Clamp	I_{IN} = -18mA		-0.8	-1.2			-1.2	
	Output voltage[1]	$\overline{CE}$ = Low							V
V_{OL}	Low	I_{OUT} = 16mA			0.45			0.5	
V_{OH}	High	I_{OUT} = -2mA	2.4			2.4			
	Input current								µA
I_{IL}	Low	V_{IN} = 0.45V			-100			-150	
I_{IH}	High	V_{IN} = 5.5V			40			50	
	Output current								
$I_{O(OFF)}$	Hi-Z state	$\overline{CE}$ = High, V_{OUT} = 0.5V			-40			-60	µA
		$\overline{CE}$ = High, V_{OUT} = 5.5V			40			60	
I_{OS}	Short circuit[3]	$\overline{CE}$ = Low, V_{OUT} = 0V High Stored	-20		-70	-15		-85	mA
I_{CC}	V_{CC} supply current			110	155			155	mA
	Capacitance	$\overline{CE}$ = High, V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V		8			8		

AC ELECTRICAL CHARACTERISTICS

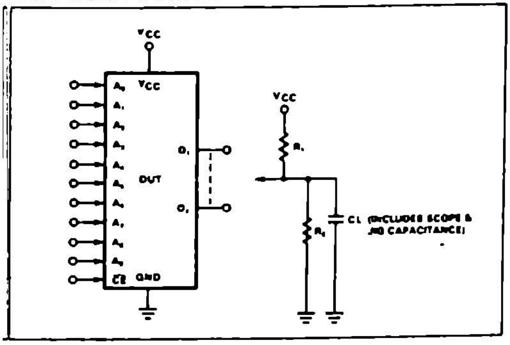
R_1 = 270Ω, R_2 = 600Ω, C_L = 30pF
N82HS185: 0°C ≤ T_A ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V
S82HS185: -55°C ≤ T_A ≤ +125°C, 4.5V ≤ V_{CC} ≤ 5.5V

PARAMETER	TO	FROM	N82HS185			S82HS185			UNIT
			Min	Typ[5]	Max	Min	Typ	Max	
Access time									ns
T_{AA}[4]	Output	Address		40	50			70	
T_{CE}	Output	Chip enable		20	30			40	
Disable time									ns
T_{CD}	Output	Chip disable		20	30			40	

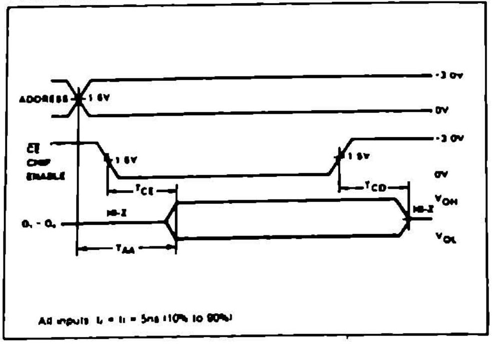
NOTES

[1] All voltage values are with respect to network ground terminal.
[2] Positive current is defined as into the terminal referenced.
[3] Duration of the short circuit should not exceed 1 second.
[4] Tested at an address cycle time of 1µsec.
[5] All typical values are at V_{CC} = 5V, T_A = 25°C.

TEST LOAD CIRCUIT


VCC
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
VCC
OUT
O1
O4
CE
GND
VCC
R1
R2
CL (INCLUDES SCOPE & JIG CAPACITANCE)

VOLTAGE WAVEFORM


ADDRESS
1.5V
+3.0V
0V
CE
CHIP ENABLE
1.5V
1.5V
+3.0V
0V
TCE
TCD
HI-Z
HI-Z
O1 - O4
VOH
VOL
TAA

All inputs t_r = t_f = 5ns (10% to 90%)

16,384-BIT BIPOLAR PROM (2048X8) — 82HS191 (T.S.)

Preliminary

82HS191-I,G,N,R

DESCRIPTION

The 82HS191 is field programmable, which means that custom patterns are immediately available by following the fusing procedure given in this data manual. The 82HS191 is supplied with all outputs at a logical low. Outputs are programmed to a logic high level at any specified address by fusing a Ni-Cr link matrix.

This device includes on-chip decoding and 3 chip enable inputs for ease of memory expansion. It features tri-state outputs for optimization of word expansion in bused organizations.

The 82HS191 device is available in the commercial and military ranges. For the commercial temperature range (0°C to +75°C) specify N82HS191, I, N, and for the military temperature range (-55°C to +125°C) specifiy S82HS191, I, R or G

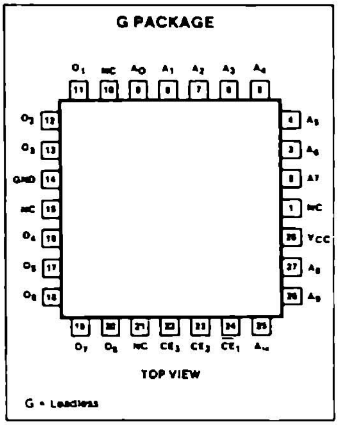
FEATURES

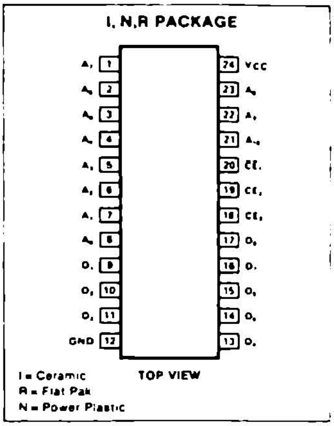
- **Address access time:**
  - **N82HS191: 60ns max**
  - **S82HS191: 80ns max**
- **Power dissipation: 20µW/bit typ**
- **Input loading:**
  - **N82HS191: -100µA max**
  - **S82HS191: -150µA max**
- **3 chip enable inputs**
- **On-chip address decoding**
- **No separate fusing pins**
- **Unprogrammed outputs are low level**
- **Fully TTL compatible**

APPLICATIONS

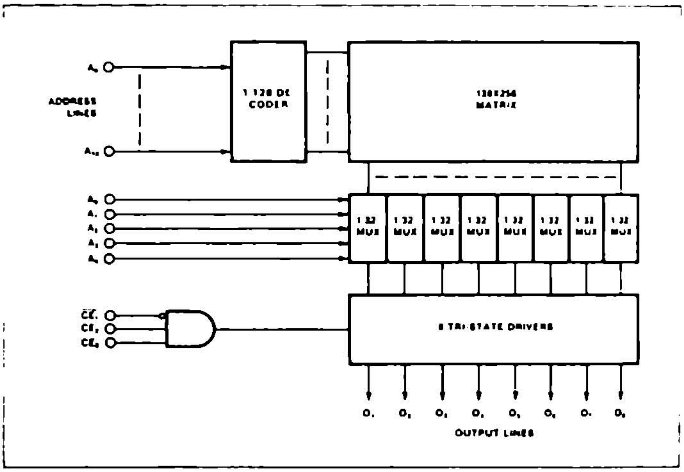
- **Prototyping/volume production**
- **Sequential controllers**
- **Microprogramming**
- **Hardwired algorithms**
- **Control store**
- **Random logic**
- **Code conversion**

PIN CONFIGURATIONS


G PACKAGE
TOP VIEW
G = Leadless


I, N,R PACKAGE
VCC
GND
TOP VIEW
I = Ceramic
R = Flat Pak
N = Power Plastic

BLOCK DIAGRAM


ADDRESS LINES
1 128 DECODER
128X256 MATRIX
1 32 MUX
8 TRI-STATE DRIVERS
OUTPUT LINES

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
Vcc	Supply voltage	+7	Vdc
VIN	Input voltage	+5.5	Vdc
	Output voltage		Vdc
Vo	Off-state	+5.5	
	Temperature range		°C
TA	Operating		
	N82HS191	0 to +75	
	S82HS191	-55 to +125	
TSTG	Storage	-65 to +150	

Preliminary

82HS191-I,G,N,R

DC ELECTRICAL CHARACTERISTICS

N82HS191: 0°C ≤ T_A ≤ -75°C, 4.75V ≤ V_{CC} ≤ 5.25V
S82HS191: -55°C ≤ T_A ≤ +125°C, 4.5V ≤ V_{CC} ≤ 5.5V

PARAMETER		TEST CONDITIONS[1]	N82HS191			S82HS191			UNIT
			Min	Typ[3]	Max	Min	Typ	Max	
	Input voltage								V
V_{IL}	Low				85			80	
V_{IH}	High		2.0			2.0			
V_{IC}	Clamp	I_{IN} = -18mA		-0.8	-1.2			-1.2	
	Output voltage	$\overline{CE}_1$ = Low, $CE_{2,3}$ = High							V
V_{OL}	Low	I_{OUT} = 9.6mA			0.45			0.5	
V_{OH}	High	I_{OUT} = -2mA	2.4			2.4			
	Input current								μA
I_{IL}	Low	V_{IN} = 0.45V			-100			-150	
I_{IH}	High	V_{IN} = 5.5V			40			50	
	Output current								μA
$I_{O(OFF)}$	Hi-Z state	$\overline{CE}_1$ = High, $CE_{2,3}$ = Low, V_{OUT} = 0.5			-40			-60	
		$\overline{CE}_1$ = High, $CE_{2,3}$ = Low, V_{OUT} = 5.5			40			60	
I_{OS}	Short circuit	$\overline{CE}_1$ = Low, $CE_{2,3}$ = High, V_{OUT} = 0V	-20		-70	-15		-85	mA
I_{CC}	V_{CC} supply current			130	175			185	mA
	Capacitance	$\overline{CE}_1$ = High, $CE_{2,3}$ = Low V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V		8			8		

AC ELECTRICAL CHARACTERISTICS

R_1 = 470Ω, R_2 = 1kΩ, C_L = 30pF
N82HS191: 0°C ≤ T_A ≤ -75°C, 4.75V ≤ V_{CC} ≤ 5.25V
S82HS191: -55°C ≤ T_A ≤ -125°C, 4.5V ≤ V_{CC} ≤ 5.5V

PARAMETER		TO	FROM	N82HS191			S82HS191			UNIT
				Min	Typ[3]	Max	Min	Typ	Max	
	Access time									ns
T_{AA}[2]		Output	Address		50	60			80	
T_{CE}		Output	Chip enable		20	35			45	
	Disable time									ns
T_{CD}		Output	Chip disable		20	35			45	

NOTES

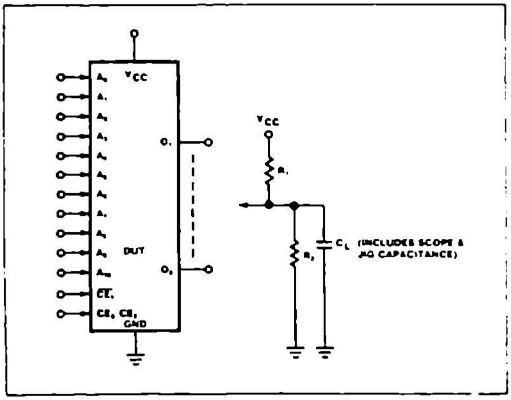
[1] Positive current is defined as into the terminal referenced

[2] Tested at an address cycle time of 1 μsec

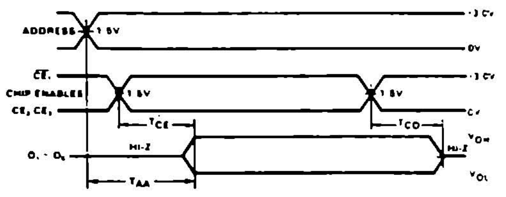
[3] All typical values are at V_{CC} = 5V, T_A = 25°C

Preliminary

TEST LOAD CIRCUIT


V_{CC}
A_0
A_1
A_2
A_3
A_4
A_5
A_6
A_7
A_8
A_9
A_{10}
$\overline{CE}_1$
CE_2, CE_3
GND
DUT
O_1
O_8
V_{CC}
R_1
R_2
C_L (INCLUDES SCOPE & JIG CAPACITANCE)

VOLTAGE WAVEFORM


ADDRESS
1.5V
+3.0V
0V
$\overline{CE}_1$
CHIP ENABLES
CE_2, CE_3
1.5V
1.5V
+3.0V
0V
T_{CE}
T_{CO}
O_1 - O_8
HI-Z
HI-Z
V_{OH}
V_{OL}
T_{AA}

All inputs: $t_R = t_F = 5$ns (10% to 90%)

16384-BIT BIPOLAR PROM (4096×4) 82HS195

Advance Information

DESCRIPTION

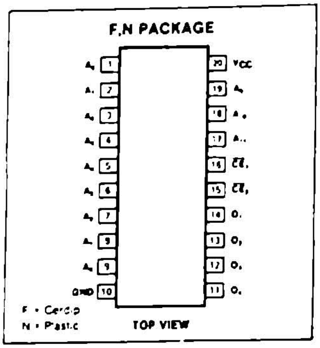
The 82S195 is field programmable, which means that custom patterns are immediately available by following the fusing procedure given in this data manual. The standard 82HS195 is supplied with all outputs at logical low. Outputs are programmed to a logic high level at any specified address by fusing a programmable link matrix.

This device includes on-chip decoding and 2 chip enable inputs for memory expansion. It features tri-state outputs for optimization of word expansion in bused organizations.

The 82HS195 device is available in the commercial and military temperature ranges. For the commercial temperature range (0°C to + 75°C) specify N82HS195, I, N, and for the military temperature range (– 55°C to + 125°C) specify S82HS195, F.

FEATURES

- **Low power dissipation: 35μW/bit typ**
- **Address access time:**
  - **N82HS195: 35ns max**
  - **S82HS195: 50ns max**
- **Input loading:**
  - **N82HS195: – 100μA max**
  - **S82HS195: – 150μA max**
- **On-chip address decoding**
- **No separate fusing pins**
- **Unprogrammed outputs are low level**
- **Fully TTL compatible**

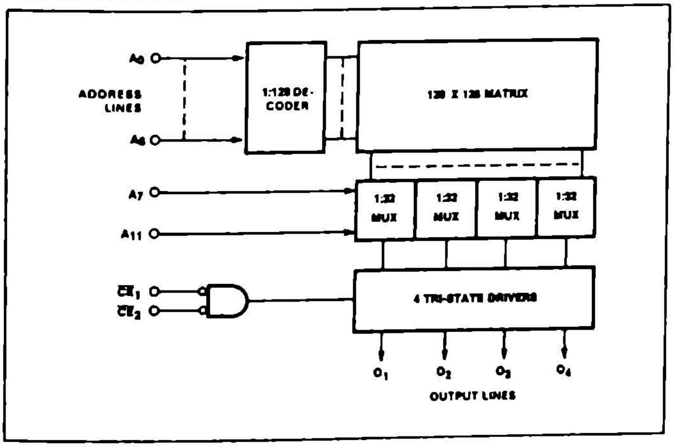

F,N PACKAGE
1
2
3
4
5
6
7
8
9
GND 10
20 VCC
19
18
17
16 CE
15 CE
14 O
13 O
12 O
11 O
TOP VIEW

F = Cerdip
N = Plastic

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Supply voltage	-7	Vdc
V_{IN}	Input voltage	-5 5	Vdc
	Output voltage		Vdc
V_O	Off-state	-5 5	
	Temperature range		°C
T_A	Operating		
	N82HS195	0 to -75	
	S82HS195	-55 to -125	
T_{STG}	Storage	-65 to -150	

BLOCK DIAGRAM


ADDRESS LINES
A0
A6
1:128 DECODER
128 X 128 MATRIX
A7
A11
1:32 MUX
1:32 MUX
1:32 MUX
1:32 MUX
CE1
CE2
4 TRI-STATE DRIVERS
O1
O2
O3
O4
OUTPUT LINES

Advance Information

82HS195-F,N

DC ELECTRICAL CHARACTERISTICS

N82HS195: 0°C $\leq T_A \leq +75$°C, 4.75V $\leq V_{CC} \leq$ 5.25V
S82HS195: −55°C $\leq T_A \leq +125$°C, 4.5V $\leq V_{CC} <$ 5.5V

	PARAMETER	TEST CONDITIONS[1,2]	N82S195			S82S195			UNIT
			Min	Typ[5]	Max	Min	Typ	Max	
	Input voltage[1]								V
V_{IL}	Low				85			80	
V_{IH}	High		2.0			2.0			
V_{IC}	Clamp	I_{IN} = -18mA		-0.8	-1.2			-1.2	
	Output voltage[1]	$\overline{CE}_1$ & $\overline{CE}_2$ = Low							V
V_{OL}	Low	I_{OUT} = 16mA			0.45			0.5	
V_{OH}	High	I_{OUT} = -2mA	2.4			2.4			
	Input current								µA
I_{IL}	Low	V_{IN} = 0.45V			-100			-150	
I_{IH}	High	V_{IN} = 5.5V			40			50	
	Output current								
$I_{O(OFF)}$	Hi-Z state	$\overline{CE}_1$ & $\overline{CE}_2$ = High, V_{OUT} = 0.5V			-40			-60	µA
		$\overline{CE}_1$ & $\overline{CE}_2$ = High, V_{OUT} = 5.5V			40			60	
I_{OS}	Short circuit[3]	$\overline{CE}_1$ & $\overline{CE}_2$ = Low, V_{OUT} = 0V High Stored	-20		-70	-15		-85	mA
I_{CC}	V_{CC} supply current			110	155			165	mA
	Capacitance	$\overline{CE}_1$ & $\overline{CE}_2$ = High, V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V		8			8		

AC ELECTRICAL CHARACTERISTICS

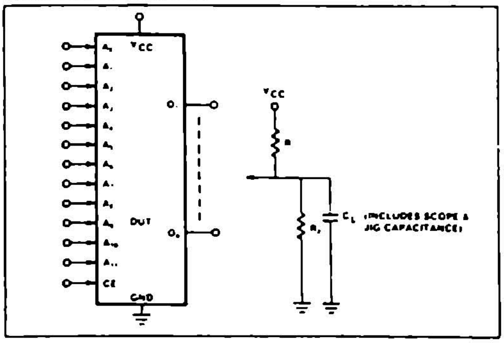
R_1 = 270Ω, R_2 = 600Ω, C_L = 30pF
N82HS195: 0°C $\leq T_A \leq +75$°C, 4.75V $\leq V_{CC} \leq$ 5.25V
S82HS195: −55°C $\leq T_A \leq +125$°C, 4.5V $\leq V_{CC} \leq$ 5.5V

PARAMETER	TO	FROM	N82S195			S82S195			UNIT
			Min	Typ[5]	Max	Min	Typ	Max	
Access time									ns
T_{AA}[4]	Output	Address			35			50	
T_{CE}	Output	Chip enable			25			30	
Disable time									ns
T_{CD}	Output	Chip disable			25			30	

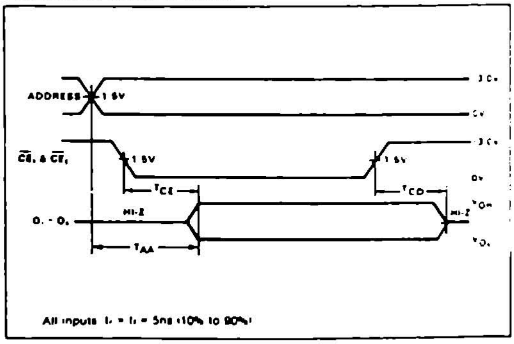
NOTES

1 All voltage values are with respect to network ground terminal.
2 Positive current is defined as into the terminal referenced
3 Duration of the short circuit should not exceed 1 second
4 Tested at an address cycle time of 1µsec
5 All typical values are at V_{CC} = 5V, T_A = 25°C

TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



All inputs: tr = tf = 5ns (10% to 90%)

VALVO

Section 8
Package Outlines

VALVO

PACKAGE OUTLINES

INTRODUCTION

The following information applies to all packages unless otherwise specified on individual package outline drawings.

General

1. Dimensions shown are metric units (millimeters), except those in parentheses which are English units (inches).
2. Lead spacing shall be measured within this zone.
   a. Shoulder and lead tip dimensions are to center-line of leads.
3. Tolerances non-cumulative.
4. Thermal resistance values are determined by utilizing the linear temperature dependence of the forward voltage drop across the substrate diode in a digital device to monitor the junction temperature rise during known power application across V_{CC} and ground. The values are based upon 120 mils square die for plastic packages and a 90 mils square die in the smallest available cavity for hermetic packages. All units were solder mounted to P.C. boards, with standard stand-off, for measurement.

Plastic Only

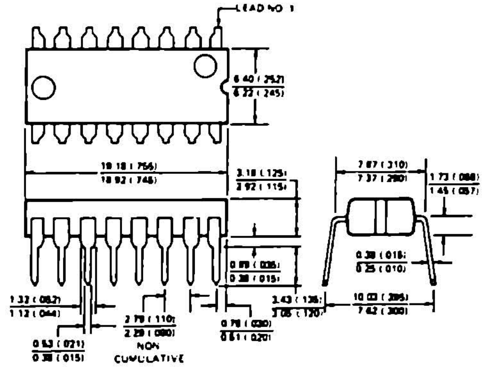
5. Lead material: Alloy 42 (Nickel/Iron Alloy) Olin 194 (Copper Alloy) or equivalents, solder dipped.
6. Body material: Plastic (Epoxy)
7. Round hole in top corner denotes lead No. 1.
8. Body dimensions do not include molding flash.

Hermetic Only

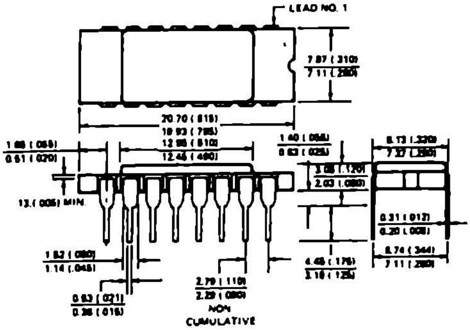
9. Lead material
   a. ASTM alloy F-15 (KOVAR) or equivalent—gold plated, tin plated, or solder dipped.
   b. ASTM alloy F-30 (Alloy 42) or equivalent—tin plated, gold plated, or solder dipped.
   c. ASTM alloy F-15 (KOVAR) or equivalent—gold plated.
10. Body Material
    a. Eyelet, ASTM alloy F-15 or equivalent—gold or tin plated, glass body.
    b. Ceramic with glass seal at leads.
    c. BeO ceramic with glass seal at leads.
    d. Ceramic with ASTM alloy F-30 or equivalent.
11. Lid Material
    a. Nickel or tin plated nickel, weld seal.
    b. Ceramic, glass seal
    c. ASTM alloy F-15 or equivalent, gold plated, alloy seal.
    d. BeO ceramic with glass seal.
12. Signetics symbol, angle cut, or lead tab denotes Lead No 1.
13. Recommended minimum offset before lead bend.
14. Maximum glass climb 0.010 inches
15. Maximum glass climb or lid skew is 0.010 inches.
16. Typical four places.
17. Dimension also applies to seating plane.

PACKAGE OUTLINES

N PACKAGE — PLASTIC
(16-PIN)

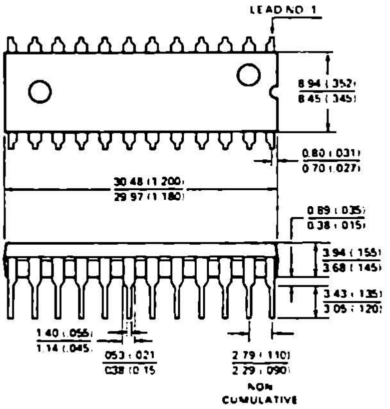

LEAD NO. 1
6.40 (.252)
6.22 (.245)
19.18 (.755)
18.92 (.745)
3.18 (.125)
2.92 (.115)
7.87 (.310)
7.37 (.290)
1.73 (.068)
1.45 (.057)
0.89 (.035)
0.38 (.015)
0.38 (.015)
0.25 (.010)
1.32 (.052)
1.12 (.044)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE
0.76 (.030)
0.51 (.020)
3.43 (.135)
3.05 (.120)
10.03 (.395)
7.62 (.300)
0.53 (.021)
0.38 (.015)

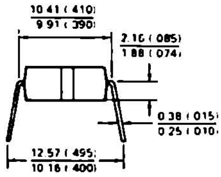
I PACKAGE — HERMETIC
(16-PIN)


LEAD NO. 1
7.87 (.310)
7.11 (.280)
20.70 (.815)
19.93 (.785)
12.95 (.510)
12.45 (.490)
1.68 (.066)
0.51 (.020)
1.40 (.055)
0.63 (.025)
8.13 (.320)
7.37 (.290)
3.05 (.120)
2.03 (.080)
13 (.005) MIN.
0.31 (.012)
0.20 (.008)
8.74 (.344)
7.11 (.280)
1.52 (.060)
1.14 (.045)
4.45 (.175)
3.18 (.125)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE
0.53 (.021)
0.38 (.015)

PACKAGE OUTLINES

N PACKAGE — PLASTIC SLIM LINE
(24-PIN)


LEAD NO 1
8.94 (.352)
8.45 (.345)
0.80 (.031)
0.70 (.027)
30.48 (1.200)
29.97 (1.180)
0.89 (.035)
0.38 (.015)
3.94 (.155)
3.68 (.145)
3.43 (.135)
3.05 (.120)
1.40 (.055)
1.14 (.045)
.053 (.021)
.038 (.015)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE

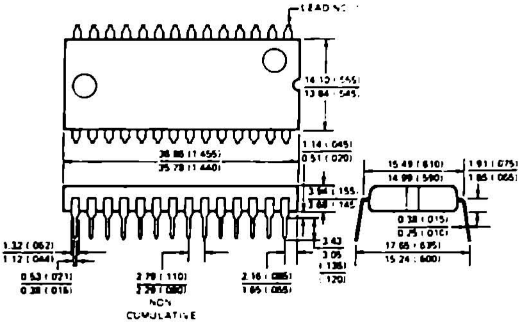

10.41 (.410)
9.91 (.390)
2.16 (.085)
1.88 (.074)
0.38 (.015)
0.25 (.010)
12.57 (.495)
10.16 (.400)

I PACKAGE — HERMETIC SLIM LINE
(24-PIN)

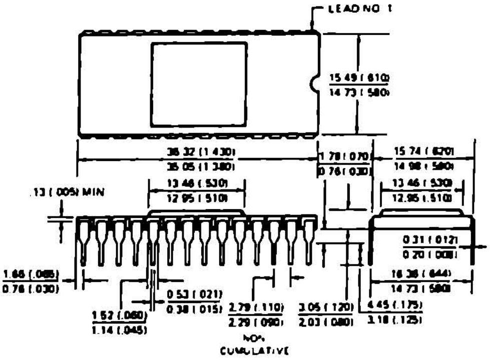
CONTACT YOUR SIGNETICS SALES OFFICE

PACKAGE OUTLINES

N PACKAGE — PLASTIC
(28-PIN)

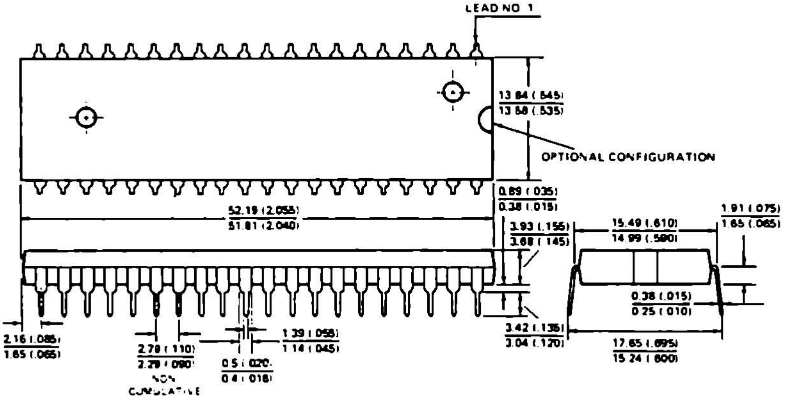

LEAD NO. 1
14.10 (.555)
13.84 (.545)
1.14 (.045)
0.51 (.020)
36.96 (1.455)
35.78 (1.440)
15.49 (.610)
14.99 (.590)
1.91 (.075)
1.65 (.065)
3.94 (.155)
3.68 (.145)
0.38 (.015)
0.25 (.010)
3.43
3.05
(.135)
(.120)
17.65 (.695)
15.24 (.600)
1.57 (.062)
1.12 (.044)
0.53 (.021)
0.38 (.015)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE
2.16 (.085)
1.65 (.065)

I PACKAGE — HERMETIC
(28-PIN)

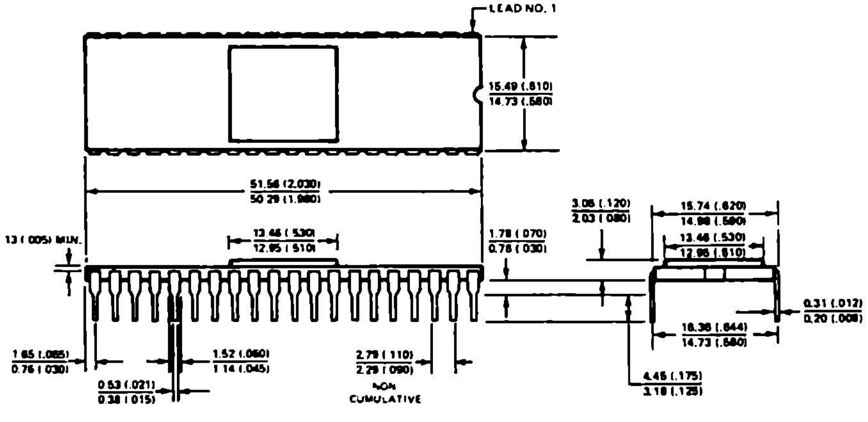

LEAD NO. 1
15.49 (.610)
14.73 (.580)
36.32 (1.430)
35.05 (1.380)
1.78 (.070)
0.76 (.030)
15.74 (.620)
14.98 (.590)
.13 (.005) MIN
13.46 (.530)
12.95 (.510)
13.46 (.530)
12.95 (.510)
0.31 (.012)
0.20 (.008)
1.66 (.065)
0.76 (.030)
16.38 (.644)
14.73 (.580)
0.53 (.021)
0.38 (.015)
1.52 (.060)
1.14 (.045)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE
3.05 (.120)
2.03 (.080)
4.45 (.175)
3.18 (.125)

PACKAGE OUTLINES

N PACKAGE — PLASTIC
(40-PIN)


LEAD NO. 1
13.84 (.545)
13.58 (.535)
OPTIONAL CONFIGURATION
0.89 (.035)
0.38 (.015)
52.19 (2.055)
51.81 (2.040)
3.93 (.155)
3.68 (.145)
15.49 (.610)
14.99 (.590)
1.91 (.075)
1.65 (.065)
0.38 (.015)
0.25 (.010)
3.42 (.135)
3.04 (.120)
17.65 (.695)
15.24 (.600)
2.16 (.085)
1.65 (.065)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE
0.5 (.020)
0.4 (.016)
1.39 (.055)
1.14 (.045)

I PACKAGE — HERMETIC
(40-PIN)


LEAD NO. 1
15.49 (.610)
14.73 (.580)
51.56 (2.030)
50.29 (1.980)
13 (.005) MIN.
13.46 (.530)
12.95 (.510)
1.78 (.070)
0.76 (.030)
3.05 (.120)
2.03 (.080)
15.74 (.620)
14.99 (.590)
13.46 (.530)
12.95 (.510)
0.31 (.012)
0.20 (.008)
16.36 (.644)
14.73 (.580)
4.45 (.175)
3.18 (.125)
1.65 (.065)
0.76 (.030)
1.52 (.060)
1.14 (.045)
0.53 (.021)
0.38 (.015)
2.79 (.110)
2.29 (.090)
NON
CUMULATIVE

Valvo Unternehmensbereich Bauelemente der Philips GmbH

Burchardstraße 19, Postfach 10 63 23, 2000 Hamburg 1
Telefon (0 40) 32 96–0, Telex 2 15 401–0 va d, Telefax (0 40) 32 96–213

Valvo Zweigbüros

Berlin/Hamburg
Burchardstraße 19
2000 Hamburg 1
Tel. (0 40) 32 96–245...248
Telex 2 15 401–65 va d

Essen
Dreilindenstraße 75–77
4300 Essen
Tel. (02 01) 23 60 01
Telex 8 571 136 siva d

Frankfurt
Theodor-Heuss-Allee 106
6000 Frankfurt/M. 90
Tel. (06 11) 7 91 33 70/71
Telex 4 12 405 valvo d

Freiburg
Tullastraße 72
7800 Freiburg
Tel. (07 61) 50 80 91
Telex 7 721 627 vav d

Hannover
Ikarusallee 1a
3000 Hannover 1
Tel. (05 11) 63 00 94
Telex 9 230 239 vav d

München
Ridlerstraße 37
8000 München 2
Tel. (0 89) 51 04–372
Telex 5 213 015 siva d

Nürnberg
Bessemerstraße 14
8500 Nürnberg 10
Tel. (09 11) 5 10 91
Telex 6 23 829 vav d

Stuttgart
Höhenstraße 21
7012 Fellbach
Tel. (07 11) 52 30 13
Telex 7 254 755 siva d

Valvo Distributoren

Berlin
distron GmbH & Co.
Behaimstraße 3
1000 Berlin 10
Tel. (0 30) 3 42 10 41
Telex 1 85 478

Bremen
Mütron, Müller GmbH & Co.
Bornstraße 22
2800 Bremen 1
Tel. (04 21) 30 56–0
Telex 2 45325

Frankfurt
Spoerle Electronic KG
Bauelemente Distributor
Max-Planck-Straße 1–3
6072 Dreieich b. Frankfurt
Tel. (0 61 03) 3 04–1
Telex 4 17 972

Göttingen
Retron GmbH
Rodeweg 18
3400 Göttingen
Tel. (05 51) 9 04–0
Telex 9 6 733

Hamburg
Walter Kluxen
Bauelemente für Elektronik
Nordkanalstraße 52
2000 Hamburg 1
Tel. (0 40) 24 89–0
Telex 2 162 074

Hannover
Elkose GmbH
Geschäftsbereich Hannover
Vahrenwalder Straße 219 A
3000 Hannover 1
Tel. (05 11) 63 99 63
Telex 9 21 501

München
Sasco GmbH
Hermann-Oberth-Straße 16
8011 Putzbrunn b. München
Tel. (0 89) 46 11–1
Telex 5 29 504

Ultratronik GmbH
Münchnerstraße 6
8031 Seefeld 1
Tel. (0 81 52) 77 73
Telex 5 26 459

Stuttgart
elecdis Ruggaber GmbH
Hertichstraße 41
7250 Leonberg-Eltingen
Tel. (0 71 52) 4 70 81
Telex 7 24 192

Elkose GmbH
Geschäftsbereich Möglingen
Bahnhofstraße 44
7141 Möglingen
Tel. (0 71 41) 4 87–1
Telex 7 264 472

Wuppertal
Herbert M. Müller
Vertriebsgesellschaft mbH
Vereinstraße 17
5600 Wuppertal 1
Tel. (02 02) 42 60 16
Telex 8 591 543

[redacted]